Optimization of Sensors Deployment in a 3D-Environment under the Coverage, Connectivity and Energy Consumption Constraints

BY

Tamer Mohamed Deyab

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

Systems Engineering

June 2011

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DEPARTMENT OF SYSTEMS ENGINEERING

This thesis, written by **Tamer Mohamed Deyab** under the direction of his thesis advisor Dr. Shokri Selim, thesis co-advisor Dr. Uthman Baroudi and approved by his thesis committee, has been presented to and accepted by the Dean of graduate studies, in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE IN SYSTEMS ENGINEERING.

4.1.2012

Dr. Shokri Selim
Thesis Advisor

U. Baroud 04/01/2012

Dr. Uthman Baroudi
Thesis Co-Advisor

04/01/2012

Dr. Malick Ndiaye
Member

m. Faizan.
04/01/2012

Dr. Faizan Mysorewala
Member

Jan 4, 2012

Dr. Ahmad Masoud
Member

Jan 7, 12

Dr. Fouad Alsunni
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

Date: 14/1/12

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Dedicated to

My family members

Parents, brothers and sisters

ACKNOWLEDGMENT

All praises are for ALLAH, the most compassionate, the merciful. May peace and blessing be upon his prophet Mohammed (PBUH), his family and his companions. I thank ALLAH (SWT) for giving me the knowledge and patience to complete this thesis.

Acknowledgment is due the king Fahd University of Petroleum and Minerals for giving me this chance to accomplish this achievement. I appreciate the stimulating and pleasant environment in the university. Special thanks and appreciation to the Systems Engineering Department.

Many people supported me during the completion of this thesis with criticism, helpful, assistance and references. I'm deeply indebted and thankful to my academic and thesis advisor, Professor *Shokri Selim*, for this guidance, encouragement and support. He acted like a father more than an academician. Many thanks also to the academic co-advisor, *Dr. Uthman Baroudi* for his guide, helpful insights and efforts. Also, I would like to thank the committee members, *Dr. Muhammad Mysorewala*, *Dr. Malick Nidaye* and *Dr. Ahmed Masoud* for their helpful instructions and kindness. Their advice and patience in this thesis and during the entire master program and the courses they taught to me is appreciated. Special thanks to Professor Shokri Selim for his valuable effort and motivation.

Thanks are due to my friends and mates for their interest and cooperation. Many thanks and appreciation to Professor *Salih Duffua* and Professor *Fouad Al-Sunni* for helping me with my admission and supporting me throughout the program. Thanks also to everyone dropped a smile or a good wish on my way. Also, I would like to thank *Mr. Nuruzzaman* for his valuable assistance in MatLab software.

Last but certainly not least, I would like to thank my family for their support, motivation and stand beside me. I would like to show my gratitude to every member of my dear family for the generous love and encouragement they gave to me. This thesis is dedicated to all of them.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

THESIS ABSTRACT

Name: TAMER MOHAMED DEYAB

Title: OPTIMIZATION OF SENSORS DEPLOYMENT IN 3D-ENVIRONMENT UNDER THE COVERAGE, CONNECTIVITY AND ENERGY CONSUMPTION CONSTRAINTS

Degree: MASTER OF SCIENCE

Major Field: SYSTEMS ENGINEERING

Date of Degree: JUNE 2011

The recent advancements in micro electronic systems and smart sensing material have enabled the manufacturing of miniature and smart wireless sensors. Sensors are used for detecting physical phenomena such as temperature, pressure, sound, light, electro-magnetic field, vibration, gas leaks, etc… In practice, the surveillance of critical areas requires the placement of sensors such that several criteria are satisfied, for example, coverage, connectivity, energy dissipation, cost…etc. This thesis deals with finding the optimal placement of wireless sensors and signal relays in 2-D or 3-D space. The relays receive signals from the sensors and re-transmit them to other relays all the way to a central processing node. A bi-objective integer linear program (ILP) is designed to find the locations of the sensors and the relays while minimizing the total network cost and the total network power consumption simultaneously. The ILP locates the sensors such that all critical points are covered by their minimum criticality. The relays are placed so that the network connectivity is guaranteed. A space partitioning heuristic method is designed to overcome the expensive computational time of the ILP and provide near-optimal solutions. Solved examples are presented to show the output and performance of both the ILP and the heuristic method.

Keywords: optimization, integer linear program, sensors, relays, wireless sensors network, space partitioning heuristic.

MASTER OF SCIENCE

KING FAHD UNIVERSITY OF PETROLUEM AND MINERALS

DHAHRAN-SAUDI ARABIA

JUNE 2011

خلاصـــــــة الأطــروحة

الاسم: تــامر محمد ديـــــــــــــاب

العنوان: التوزيع الامثل للمجسات اللاسلكية في البيئة ثلاثية الابعاد باعتبار التغطية و الاتصال و استهلاك الطاقة

الدرجة: ماجستير في العلوم

التخصص: هندسة النظم

التاريخ: يونيو 2011

التطورات الحديثة في مجال الانظمة الالكترونية الدقيقة و المجسات الرقمية ساهمت في صنع مجسات لاسلكية صغيرة الحجم و ذكية. تستخدم المجسات في مجال مراقبة المتغيرات الفيزيائية كدرجة الحرارة و الضغط و الصوت و الضوء و المجال المغنطيسي و الاهتزازات و تسربات السوائل و الغاز في المناجم. من اجل مراقبة المواقع الحساسة يجب وضع المجسات بحيث تضمن بعض العناصر المهمة مثل التغطية و الاتصال و استهلاك الطاقة.

تركز هذه الاطروحة على ايجاد المواقع المثلى للمتجسات و قواعد الاتصال البينية في بيئة ثنائية او ثلاثية الابعاد. تستخدم قواعد الاتصال البينية لاستقبال الاشارات من المتجسات و ارسالها الى قواعد الاتصال البينية اخرى حتى تصل الاشارات الى مركز المعالجة. و قد تم بناء برنامج خطي مزدوج الاهداف لايجاد مواقع المتجسات و قواعد الاتصال البينية بحيث تضمن أقل تكلفة اجمالية للمجسات و قواعد الاتصال البينية و تقلل استهلاك الطاقة الى ادنى مستوى يمكن بلوغه.

يقوم البرنامج الخطي بتحديد مواقع المتجسات بحيث يتم تغطية كل موقع حساس بعدد من المجسات يتناسب مع اهمية الموقع. و يتم ايضا تحديد مواقع قواعد الاتصال البينية لضمان اتصالية الشبكة و قدرة نقل الاشارات لمركز المعالجة. لتجنب الوقت الكبير الذي يستهلكه البرنامج الخطي لايجاد المواقع فقد تم تصميم طريقة مبتكرة لايجاد حل شبه مثالي في وقت سريع. ختمت الاطروحة بتقديم توصيات و مقترحات للبحوث المستقبلية في هذا المجال.

درجة الماجستير في العلوم

جامعة الملك فهد للبترول و المعادن

الظهران- المملكة العربية السعودية

يونيو 2011

Chapter 1

INTRODUCTION

1.1 PREFACE

The recent advancements in micro electronic systems and smart sensing material have enabled the manufacturing of miniature and smart wireless sensors. Sensors are used for detecting physical phenomena such as temperature, pressure, sound, light, electro-magnetic field, vibration, gas leaks, etc… In practice, the surveillance of critical areas requires the placement of sensors such that several criteria are satisfied, for example, connectivity, coverage, cost, etc. The optimization of sensors deployment problem has attracted considerable attention in recent years. In constructing such networks one is interested in minimizing the underlying cost by deciding the best placement locations for the deployed sensors. The network should retain a sufficient level of coverage for the critical locations.

In this work, we will consider a bounded area that contains critical facilities. Each facility will be represented by a point and will be called a critical point. A critical point requires one or more sensors to detect a specific phenomenon. Redundant sensors may be needed to satisfy the critical point's criticality and increase the reliability of the measurements. Sensors measure the observed phenomenon and transmit the detected signals to relays. On the other hand, relays are responsible for receiving the signals from sensors and transmitting them to a known central location called the processing node. The signal can reach the processing node through a single relay (single-hop) or through multiple relays (multi-hop). Relays are maintained to keep the network connected. We assume that there are several types of sensors, which differ in cost, sensing range and transmission range. Also, different types of relays could be available with different costs which are proportional to the transmission ranges.

We discretize the field that contains the critical points using a regular grid such that the critical points fall on the grid points. The sensors and relays will be placed on the grid points. The function of the relays in the network is to route sensors' signals to the processing node. Such routing problems are known to be NP-complete in general.

We will consider a 3D-environment in which M critical points need to be monitored. Each critical point c where, $c = 1,...M$ has a minimum requirement of the number of sensors that need to cover it. We will refer to this number as the criticality of critical point c. This criticality of critical point c, D_c, is an input to the model and it represents the significance level of the monitored point. An important assumption also is that two general types of devices would be available, Sensing devices and Relay devices. Sensing devices (i.e. sensors) can only be used to detect the observed phenomenon and send the readings to nearby relays. While, relay devices (i.e. relays) are used only to maintain a connected network through receiving and transmission of the sensors' signals. Relays play the role of an intermediate agent between sensors and the processing node by connecting a sensor and a relay or two relays.

The work in [5] studied the problem of minimum-cost sensor placement on a bounded 3D sensing field. They assumed different types of sensors are available with different sensing ranges and costs and presented a polynomial-time approximation algorithm for this problem. This work is close to our work in this paper. However, our work differs in several aspects. First, our proposed problem is more general where we assume two types of nodes (i.e. sensors and relays). Therefore, the work in [5] is a special case of ours. Second, in [5] it is assumed that the whole area needs to be covered, but in our model, we are interested with covering specific areas. Also, our model determines the position of sensors and relays as well as the signals' transmission path to the processing node for each detected signal. Thus, our model gives an overall solution detailed network design.

1.2 PROBLEM STATEMENT

A detailed literature review of the existing related work reveals that there is no well-designed model for placing sensors and relays under the coverage, connectivity and energy consumption constraints. In this thesis, we consider a scenario where a bounded space exists with critical points to be monitored by sensors. The location of the processing node, locations and criticalities of the critical points and parameters of the available sensors and relays are assumed to be known. Our designed model is aimed to find the optimal type and location of sensors (to guarantee coverage of critical points) and of relays (to guarantee connectivity to the processing node) that will minimize the total cost of the deployed network and the total power consumption simultaneously.

1.3 THESIS ORGANIZATION

The rest of the thesis is organized as follows. Chapter 2 summarizes a comprehensive survey of the literature on related work. Chapter 3 presents the designed Integer Linear Program (ILP), its assumptions, parameters and decision variables. Also, numerical examples are presented. Chapter 4 presents the designed heuristic method with solved examples. Also, a comparison is made between the ILP and the heuristic method solutions. In Chapter 5, a summary of the thesis is provided along with recommended future studies and extensions.

Chapter 2

LITERATURE REVIEW

2.1 PREFACE

In this chapter a literature survey on relevant previous work is presented. The survey starts by discussing the paper work, the targeted problem, the approach used to solve that problem and the difficulties or issues if found.

2.2 RELATED WORK

A novel grid coverage strategy for effective surveillance and target location in a distributed sensor networks was proposed in [1]. They presented the sensor's field (two or three dimensional) as a grid points where sensors could be deployed. They assumed the availability of two different sensors with different sensing range and cost. An ILP model was formulated with the goal of minimizing the total cost of sensors that maintain a full coverage of the sensor field. In 2009, the authors in [2] studied the hierarchical relay node based networks where a mobile data collector moves along a fixed trajectory to collect data from relay nodes and deliver it to the base station. This strategy of using the mobile data collector prevents the relay nodes from sending to the base station through long distances. Hence, this way increases the overall lifetime of the network by minimizing the energy dissipation. The authors presented an ILP formulation to determine an optimal relay node placement scheme taking into consideration the sensor data rates, the relay nodes buffer size and the speed of the mobile data collector. The model's objective function is to minimize the total number of relay nodes required while maintaining the buffer capacity and the maximum energy dissipation constraints. The network performance was investigated under different design parameters, such as the buffer size and the speed of the mobile data collector. To maximize the network lifetime under the constraints of coverage and connectivity, the authors in [3] constructed an Integer Linear Program model to identify the optimal allocation of sensors' states that will minimize the energy dissipation and satisfy the network requirements. In their paper, the authors assumed that the sensor can be turned on, turned off or promoted cluster head, and that a different power consumption level is associated with each of these states. The study proved that the Integer Linear Programming model is NP-Complete, mainly due to the spanning tree connectivity constraint, and they proposed a Tabu search heuristic and simulation techniques to solve the large size networks in reasonable amount of time. The work in [4] discussed an effective placement of sensors to guarantee coverage and surveillance in distributed sensor networks. In their model, the probabilistic nature of the coverage range and precision was inherently considered to determine the minimum number of sensors deployed to provide sufficient coverage of the sensor field. The authors included the issue of preferential coverage of grid points, due to their importance, in the model as well. They constructed a polynomial-time algorithm to optimize the number of sensors and determine their placement in a grid sensor field. The algorithm was applied experimentally on an example sensor field with obstacles to demonstrate its practical application. The authors in [5] studied the minimum-cost sensor placement in a bounded 3D sensing field. In their study, the authors proposed the availability of different types of sensors with different sensing ranges and different costs. The problem then becomes to find a selection of sensors and subset of points such that every point in the bounded space is covered by at least a specified lower bound, $\sigma \geq 1$, of sensors and the total cost of sensors is minimized. This problem is NP-hard and the authors presented in their paper a polynomial-time approximation algorithm for this problem with some proven approximation ratio.

In [6], the authors discussed two schemes for routing and placement of mobile data collectors in Underwater Acoustic Sensor Networks. The first scheme, called the Delay Tolerant Placement and Routing (DTPR), maximizes the network lifetime without any delay considerations. The second scheme, called the Delay Constrained Placement and Routing (DCPR), maximizes the network lifetime with an upper bound on the maximum delay. Both schemes are designed for 3D environments where on-the-surface data collectors gather data from underwater sensors and relay them to the sink node. The authors presented an ILP model for both problems. Experimental analysis shows that their schemes prolong the lifetime of the network significantly when compared to other data collector placement schemes. The study in [7] focuses on maximizing the lifetime of a data flow taking into consideration the energy of each node involved in the data flow. The objective of the mathematical model is to find a tradeoff between the length of routes and the number of nodes in each route in order to achieve two objectives, the maximization of the shortest node’s lifetime and the convergence of all nodes’ lifetime to a unique value. Their model allows finding the optimal placement of sensors when they have different levels of residual energy. The study compared the proposed “energy spaced” approach to the “random” and “evenly spaced” approaches and the results showed that the suggested approach yields a longer lifetime compared to other considered approaches. The study showed that the optimal placement is on the straight line between the source and the destination, but the sensors should be spaced on that line based on their residual energies.

The authors in [8] presented the problem of placement of relay node in heterogeneous wireless sensor networks. They formulated a general node placement optimization model with the objective of minimizing the network cost subject to constraints on the lifetime and connectivity of the network. Optimal and heuristic solutions are proposed for two representative design scenarios. The first scenario is where the sensor nodes are energy constrained but relay nodes are not. This scenario has been tackled using the minimum set covering problem, [9]. The second scenario is where both of the sensor nodes and relay nodes are energy limited. For this challenging scenario, a locally optimal two-phase placement heuristics are used to the overall optimal solution. The paper in [10] dealt with placement of sensors, relays and base stations such that coverage, connectivity and routing are guaranteed. The authors proposed several placement strategies for different models’ objectives such as minimizing the number of sensor nodes deployed, minimizing the total cost, minimizing the energy consumption, maximizing the network lifetime and maximizing the network utilization. The authors considered and formulated both scenarios where the detection ranges are reliable and probabilistic as integer linear programs. The effective and performance of the proposed models are verified through simulation experiments. The authors stated that the proposed integer linear programs are NP-hard. Although some relaxation techniques could be used to obtain a good feasible solution, it worth spending time and effort to get the optimal solution in order to assess in developing polynomial time heuristic algorithms for large size problems. A major contribution of our proposed model over the author's models is that our proposed model determines the wireless transmission links and they are vital part of the final solution. In [11], the authors presented a model to utilize the deployment of mobile sensors by balancing the energy consumption and increasing the network lifetime. In their work, the authors allowed for a predetermined number of sensors to move for the purpose of covering the vacancies. The sensors were restricted to move in the disk-based mobility model. Using the improved particle swarm optimization algorithm, the authors attempted to improve the k-coverage of the mobile sensor networks. From the simulation results, they concluded that few mobile sensors are required to realize the k-coverage which results in a low cost for the sensor networks and the energy consumption in the mobile sensors.

Chapter 3

INTEGER LINEAR PROGRAM

3.1 PREFACE

In our problem, we assume that we are given a bounded space which could be either 2-dimensional or 3-dimensional. Also, the space contains M critical points which need to be continuously covered by sensors. We assume that each detected signal needs to be routed through relays only until it reaches a processing node where it can be processed and analyzed for action taking.

3.1.1 INTEGER LINEAR PROGRAM ASSUMPTIONS

The assumptions on which the ILP model is based are:

1. The field of study is 2D or 3D bounded space.
2. The field of study is partitioned into regular equally-spaced grid points. The distances in this study are taken to be of one unit length apart.

3. Sensors and relays cannot be placed on the critical points' or the processing node's location.
4. The space contains M critical points. The criticality of each critical point is reflected by the minimum number of sensors that is needed to cover it, D_i. Where $i = 1,...,M$.
5. There are N_S types of sensors. Each type is associated with a sensing range S_i and transmission range of T_{Si} and costs C_{Si}, where $i = 1,...,N_S$.
6. There are N_R types of relays. Each type is associated with a transmission range of T_{Ri} and costs C_{Ri}, where $i = 1,...,N_R$.
7. Sensors do not receive signals from other sensors or relays. They just sense the desired phenomenon and send the detected signal to a nearby relay.
8. Relays are used solely to transmit the received signal from the sensors to other relays until it reaches the processing node ultimately. Relays do not sense or detect the environment changes; they are just used as a media to help transferring the data to the processing node. Relays are needed to guarantee the connectivity of the wireless sensors network.
9. The processing node is located in a fixed location which is known in advance.
10. Data aggregation is assumed for relays. Which means that each relay waits to collect all the signals then some analysis might be done before forwarding the data to the next relay or to the processing node. This assumption validates our mathematical definition for the consumed energy. If data aggregation is not assumed, our proposed mathematical definition should be modified.

3.1.2 INTEGER LINEAR PROGRAM PARAMETERS

1. The number of grid points in the space is N_G.
2. The position coordinates (x_i, y_i, z_i), and the criticality D_i for each critical point i, where $i = 1,...,M$.
3. The predefined set U which is the set of grid points that are occupied by the critical points and the processing node. I.e. this is a set of points on which it is prohibited to place sensors or relays.
4. The Euclidean distance between each pair of grid points i and j, is $d(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$.
5. $p_{kij} = \begin{cases} 1, \text{ If a sensor of type } k \text{ placed at grid point } i \text{ covers grid point } j \\ 0, \text{ Otherwise} \end{cases}$
6. The maximum sensor transmission range, $T_{S_{\max}} = \max\left\{T_{S1}, T_{S2}, ..., T_{SN_S}\right\}$
7. The maximum relay transmission range, $T_{R_{\max}} = \max\left\{T_{R1}, T_{R2}, ..., T_{RN_R}\right\}$
8. The maximum overall transmission range, $T_{\max} = \max(T_{R_{\max}}, T_{S_{\max}})$
9. The maximum number of signals a relay can receive, V.

3.1.3 INTEGER LINEAR PROGRAM DECISION VARIABLES

1. $X_{ki} = \begin{cases} 1, \text{ if a sensor of type } k \text{ is placed at grid point } i \\ 0, \text{ Otherwise} \end{cases}$

2. $Y_{ki} = \begin{cases} 1, & \text{if a relay of type } k \text{ is placed at grid point } i \\ 0, & \text{Otherwise} \end{cases}$

3. $L_{ij} = \begin{cases} 1, & \text{if a transmission link is established between grid points } i \text{ and } j \\ 0, & \text{Otherwise} \end{cases}$

3.2 MODEL I: NETWORK TOTAL COST MINIMIZATION SUBJECT TO COVERAGE AND CONNECTIVITY CONSTRAINTS

This model's objective is to minimize the network cost, i.e. cost of sensors and relays, without considering the total energy consumption in the network. This means that the model will attempt to place the least number of sensors and relays that will satisfy the constraints of coverage and connectivity and yield the minimum possible total cost. The Integer Linear Program is described as follows:

3.2.1 THE OBJECTIVE FUNCTION FOR MODEL I

Minimize the total cost of sensors and relays $= OF_I = \sum_{k=1}^{N_S}\sum_{i=1}^{N_G} C_{Sk} X_{ki} + \sum_{k=1}^{N_R}\sum_{i=1}^{N_G} C_{Rk} Y_{ki}$

3.2.2 THE CONSTRAINTS FOR MODEL I

1. Criticality constraint:

$$\sum_{k=1}^{N_S}\sum_{i=1}^{N_G} p_{kij} X_{ki} \geq D_j \quad ; \qquad j = 1, \ldots, M \tag{3.1}$$

2. Capacity constraint:

$$\sum_{k=1}^{N_S} X_{ki} + \sum_{k=1}^{N_R} Y_{ki} \leq 1 \quad ; \qquad i = 1, \ldots, N_G \tag{3.2}$$

3. At least, one relay should be connected to the sink node:

$$\left(\sum_{k=1}^{N_R} \sum_{\substack{\forall j \\ d(j,PN) \leq T_{R_k}}} Y_{kj} \right) \geq 1 \tag{3.3}$$

4. Sensor and relays cannot be located at the grid points in the set U:

$$\sum_{j\in U}\left(\sum_{k=1}^{N_R} Y_{kj} + \sum_{k=1}^{N_S} X_{kj}\right) = 0 \tag{3.4}$$

5. At least, one arc should be connected to the sink node:

$$\sum_{k=1}^{N_R} \sum_{\substack{\forall j \\ d(j,i)\le T_{Rk}}} L_{ij} \ge 1 \quad ; \qquad i = PN \tag{3.5}$$

6. If a relay is placed at a grid point, at least one arc should be directed from that point (to allow for signal reception by that relay node):

$$\sum_{k=1}^{N_R} Y_{ki} \le \sum_{\substack{\forall j \\ d(j,PN)>d(i,PN) \\ j\ne i}} L_{ij} \quad ; \qquad i = 1,\ldots,N_G; \quad i \ne PN \tag{3.6}$$

7. If at least one arc is leaving a point, a relay must be placed at that point:

$$V\left(\sum_{k=1}^{N_R} Y_{ki}\right) \ge \sum_{\substack{\forall j \\ d(j,i)\le T_{\max} \\ d(j,PN)>d(i,PN) \\ j\ne i}} L_{ij} \quad ; \qquad i = 1,\ldots,N_G; \quad i \ne PN \tag{3.7}$$

8. If a relay is placed at point i, then at least one feasible arc will be going in to that point to allow for signal transmission by that relay:

$$Y_{ki} \le \sum_{\substack{\forall j \\ d(j,i)\le T_{Rk} \\ d(j,PN)<d(i,PN) \\ j\ne i}} L_{ji} \quad ; \qquad k = 1,\ldots,N_R; \quad i = 1,\ldots,N_G; \quad i \ne PN \tag{3.8}$$

9. If a sensor is placed at a point i, at least one feasible arc will be going in to that point to allow for signal transmission by that sensor:

$$X_{ki} \le \sum_{\substack{\forall j \\ d(j,i)\le T_{Sk} \\ d(j,PN)<d(i,PN) \\ j\ne i}} L_{ji} \quad ; \quad k=1,\dots,N_S; \quad i=1,\dots,N_G; \quad i \ne PN \tag{3.9}$$

10. The total number of transmission arcs equals the total number of sensors and relays (each sensor or relay is transmitting the received signal to one destination only):

$$\sum_{i=1}^{N_G} \sum_{\substack{j=1 \\ j\ne i \\ d(j,i)\le T_{\max}}}^{N_G} L_{ij} = \sum_{k=1}^{N_S} \sum_{\substack{i=1 \\ i\ne PN}}^{N_G} X_{ki} + \sum_{k=1}^{N_R} \sum_{\substack{i=1 \\ i\ne PN}}^{N_G} Y_{ki} \tag{3.10}$$

11. All the decision variables i.e. X_{ij}, Y_{kj} and L_{jj} are binary variables.

where: $\quad i=1,\dots,N_S; \quad k=1,\dots,N_R; \quad j=1,\dots,N_G \quad$ (3.11)

3.3 MODEL II: TOTAL ENERGY CONSUMPTION MINIMIZATION SUBJECT TO COVERAGE AND CONNECTIVITY CONSTRAINTS

This model is concerned about minimizing the total network energy consumption without considering the network cost. The total energy consumed in the network is composed of the signals transmission energy and the signals reception energy. The energy consumed in transmission of signals is assumed to be proportional to the transmission distance. i.e., signals sent to nearby destinations (relays or the processing node) will require much less energy than signals that need to be sent for longer distances. Based on that, this model will attempt to place more number of sensors and relays than what would be obtained from Model I. This will relieve the sensors and relays from the burden of sending their signals for long distances and thus minimizes their power consumption. The Integer Linear Program is described as follows:

3.3.1 THE OBJECTIVE FUNCTION FOR MODEL II

Minimize the total transmission and reception power consumption

$$OF_{II} = \sum_{i=1}^{N_G} \sum_{\substack{j=1 \\ j \neq i \\ d(j,i) \leq T_{\max}}}^{N_G} L_{ij}\left(kE_{elec} + kE_{amp} d_{ij}^2\right) + \sum_{\substack{i=1 \\ i \neq PN}}^{N_G} \sum_{\substack{j=1 \\ j \neq PN \\ j \neq i \\ d(j,i) \leq T_{\max}}}^{N_G} L_{ij}\left(kE_{elec}\right)$$

Where the first part of the objective function represents the total network transmission energy and the second part represents the total network reception energy.

3.3.2 THE CONSTRAINTS FOR MODEL II

1. All constraints in Model I except constraint (3.10).

2. Signal flow conservation constraint:

$$\sum_{\substack{\forall j \\ d(j,i)\le T_{\max} \\ d(j,PN)>d(i,PN) \\ j\neq i}} L_{ij} - \sum_{\substack{\forall j \\ d(j,i)\le T_{\max} \\ d(j,PN)<d(i,PN) \\ j\neq i}} L_{ji} \ge -\sum_{k=1}^{N_S} X_{ki} \quad ; \quad i=1,\dots,N_G; \quad i\neq PN \tag{3.12}$$

3.4 MODEL III (BI-OBJECTIVE MODEL): MINIMIZATION OF NETWORK COST AND TOTAL ENERGY CONSUMPTION

This model is designed to combine the two objectives of minimizing the network cost and the network energy consumption simultaneously. The two objectives are conflicting. i.e., the objective of minimizing the network cost will attempt to reduce the number of sensors and relays in the network while the objective of minimizing the total network energy consumption will attempt to increase their numbers. To minimize the total energy consumption in the network, i.e. transmission energy and reception energy, the following strategy will be used. First, Model I will be solved to find the minimum possible network cost which will be stored in the objective value, OF_I . This value of the network cost will be used as an upper bound for the total network cost in Model III. Then Model III will be solved with the objective of minimizing the total power consumption using all the constraints in Model II in addition to the network cost upper bound constraint. After that, the user will have the bi-objective solution for the first iteration, $\left(OF_I^1, OF_{III}^1\right)$. Then, the user will increment the cost and solve Model III to find the bi-objective solution in each of the subsequent iterations. Ultimately, a list of bi-objective solutions will be formed from which a suitable solution will be chosen. It is worthy to mention that the initial bi-objective solution will have the minimum network cost and the maximum total energy consumption. As the number of iterations increases, the network cost will increase and the total energy consumption will decrease.

The Integer Linear Program and its algorithm are described below:

3.4.1 THE OBJECTIVE FUNCTION FOR MODEL III

Minimize the total transmission and reception power consumption

$$OF_{III} = E_{Tx} + E_{Rx} = \sum_{i=1}^{N_G} \sum_{\substack{j=1 \\ j \neq i \\ d(j,i) \leq T_{\max}}}^{N_G} L_{ij}\left(kE_{elec} + kE_{amp} d_{ij}^2\right) + \sum_{\substack{i=1 \\ i \neq PN}}^{N_G} \sum_{\substack{j=1 \\ j \neq PN \\ j \neq i \\ d(j,i) \leq T_{\max}}}^{N_G} L_{ij}\left(kE_{elec}\right)$$

3.4.2 THE CONSTRAINTS FOR MODEL III

1. All constraints in Model II.
2. The total network cost upper bound constraint:

$$\sum_{k=1}^{N_S} \sum_{i=1}^{N_G} C_{Sk} X_{ki} + \sum_{k=1}^{N_R} \sum_{i=1}^{N_G} C_{Rk} Y_{ki} \leq OF_I \quad (3.13)$$

3.5 ALGORITHM A, FOR SOLVING THE BI-OBJECTIVE MODEL (MODEL III)

As described previously, this algorithm is designed to combine the two objectives of minimizing the network cost and the network energy consumption simultaneously. The two objectives are conflicting. i.e., the objective of minimizing the network cost will attempt to reduce the number of sensors and relays in the network while the objective of minimizing the energy consumption will attempt to increase their number.

The method will start by finding the least energy consumption network that corresponds to the minimum network cost. Then, the minimum network cost will be incremented in each subsequent iteration and the associated energy consumption will decrease as a result. Ultimately, the method will yield an array of bi-objective solutions for each iteration from which the user will chose an appropriate solution based on the application.

3.5.1 INITIALIZATION PROCEDURE

Step 1: Solve Model I for minimizing the total network cost. Let OF_I^1 be the optimal objective function. Denote the optimal objective value by OF_I^1.

Step 2: Solve Model II to find the minimum Total Network Energy Consumption with no limit on the Total Network Cost. Store the objective value in the variable $OF_{III}{}^*$.

Step 3: For the first iteration, $i = 1$, set $OF_I{}^* = OF_I^1$ and solve Model III by substituting OF_I^1 on the right-hand side of constraint 3.13 in Model III to determine $OF_{III}{}^1$. The bi-objective solution for the first iteration is $OF^1 = \left(OF_I^1, OF_{III}^1\right)$. This will be the first efficient point.

3.5.2 ITERATIVE PROCEDURE

Step 4: For iteration i, if the following condition holds:

$$\left(\frac{OF_{III}{}^i - OF_{III}{}^*}{OF_{III}{}^*}\right) \times 100 \leq \tau \tag{A4}$$

STOP the algorithm, otherwise go to Step 5.

Where τ is an acceptable percentage error or deviation of the current total energy value from the minimum total energy value.

Step 5: Set $i = i + 1$.

Step 6: For iteration i, set $OF_I^{\,i} = OF_I^{\,i-1} + k$ and solve Model III by substituting $OF_I^{\,i}$ on the right-hand side of constraint 3.13 in Model III to determine $OF_{III}^{\,i}$. Where k is a pre-selected increment. The bi-objective solution of iteration i is $OF^i = \left(OF_I^i, OF_{III}^i\right)$. Go to Step 4.

When the algorithm terminates, the decision maker will be left with a list of efficient points. The set of solutions obtained by the algorithm can be graphed in a form called a Pareto Chart that shows the decrease in total network energy as the network cost increases. It is the decision maker's responsibility to select an appropriate solution based on the application and severity of the monitored area. For example, if the monitored area is very critical, the decision maker might ignore the network cost and focus on the total energy consumption in order to maximize the network life time. So, in this case, the decision maker will probably pick a solution that corresponds to one of the last iterations (since the network cost goes up and the total network energy consumption goes down as the iteration number increases).

The appropriate choice of the value of k is very critical to the accuracy of the obtained solution. Choosing large value for k would lead to missing some potential points on the Pareto Chart. Also, selecting a very small value for k will lead to greater number of iterations from which some could not lead to any improvement, i.e. reduction in the total network energy consumption.

Note: The minimum value for k that will guarantee not missing any point on the Pareto Chart will be the Largest Common Divisor of the costs of all sensors and relays.

Note: Unlike the Linear Program Model, our Integer Program Model might not yield a convex function Pareto Chart. This means that some solution points might not be efficient.

3.6 NUMERICAL EXAMPLE 1 (EXACT METHOD)

Given a 2D field of 10×10 unit length dimensions. The processing node is fixed at the location $(5,5)$. Ten critical points need to be covered with different levels of criticality. The critical points' location and criticality are given in Table 3-1. We assume the availability of two types of sensors and two types of relays in the market. Table 3-2 and Table 3-3 state the characteristics of each type of sensor and each type of relay respectively.

Critical Point, i	1	2	3	4	5	6	7	8	9	10
Location Coordinates, (x_i, y_i)	(0,0)	(7,0)	(10,0)	(6,2)	(0,3)	(9,7)	(1,8)	(5,9)	(0,10)	(10,10)
Criticality, D_i	1	1	1	1	1	1	1	1	1	1

Table 3-1: Critical points' locations and criticalities for Example 1

The energy consumed during transmission of k bits for a distance of d, E_{Tx}, and the energy consumed to receive k bits, E_{Rx}, are calculated as follows:

$$E_{Tx} = kE_{elec} + kE_{amp}d^2$$
$$E_{Rx} = kE_{elec}$$

Where E_{elec} represents the electronics energy and E_{amp} represents the amplifier energy. In our example, the following values will be assumed:

$E_{elec} = 50\text{nJ/bit} = 50 \times 10^{-9}\,\text{J/bit}$

$E_{amp} = 100\text{pJ/bit/m}^2 = 100 \times 10^{-12}\,\text{J/bit/m}^2$

$k = 800$ bytes, which represents the data packet size.

Sensor's Characteristics	Sensor Type	
	S_1	S_2
Sensing Range, S_{si} (unit length)	1	2
Transmission Range, T_{si} (unit length)	1	1
Unit Cost, C_{si} ($)	2	3

Table 3-2: Two types of sensors with their respective characteristics

Relay's Characteristics	Relay Type	
	R_1	R_2
Transmission Range, T_{Ri} (unit length)	2	4
Unit Cost, C_{Ri} ($)	2	3

Table 3-3: Two types of relays with their respective characteristics

The results and of the Exact Method are summarized in the following table (the cost is in dollars and the total energy is in milli-joules):

	ILP Solution	
Iteration	OF_I **= Network Cost ($)**	OF_{III} **= Total Energy Consumption (mJ)**
First Iteration, OF^1	$OF_I^1 = 33$	$OF_{II}^1 = 0.96616$
Second Iteration, OF^2	$OF_I^2 = 34$	$OF_{II}^2 = 0.96568$
Third Iteration, OF^3	$OF_I^3 = 35$	$OF_{II}^3 = 0.96568$
Fourth Iteration, OF^4	$OF_I^4 = 36$	$OF_{II}^4 = 0.9656$
Fifth Iteration, OF^5	$OF_I^5 = 37$	$OF_{II}^5 = 0.9656$
Sixth Iteration, OF^6	$OF_I^6 = 38$	$OF_{II}^6 = 0.9656$
Seventh Iteration, OF^7	$OF_I^7 = 39$	$OF_{II}^7 = 0.9656$

Table 3-4: Exact method's solution for example 1

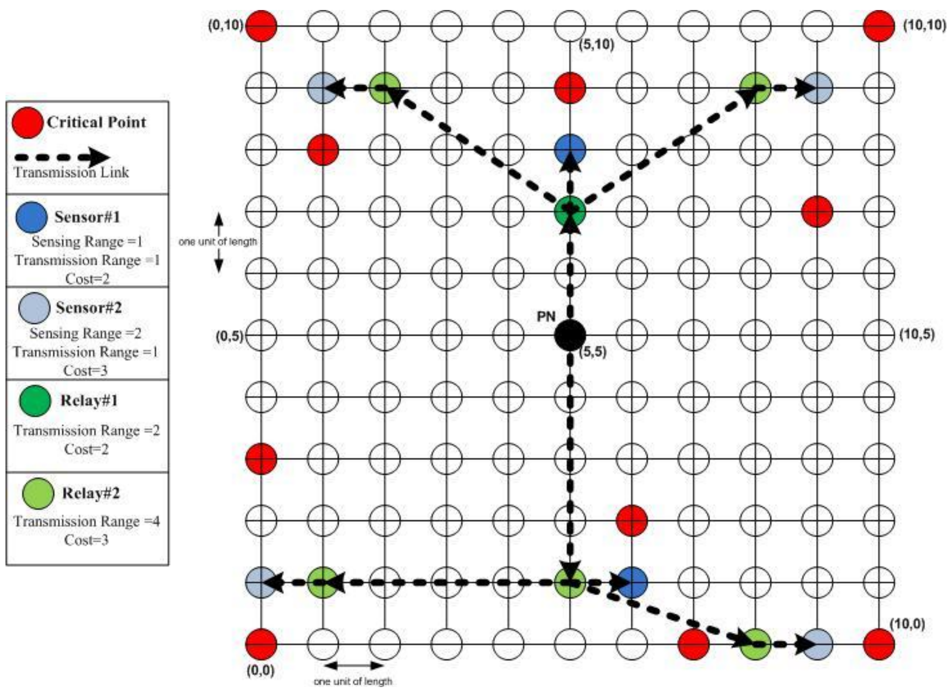

Figure 3-1: Model I solution for Iteration 1

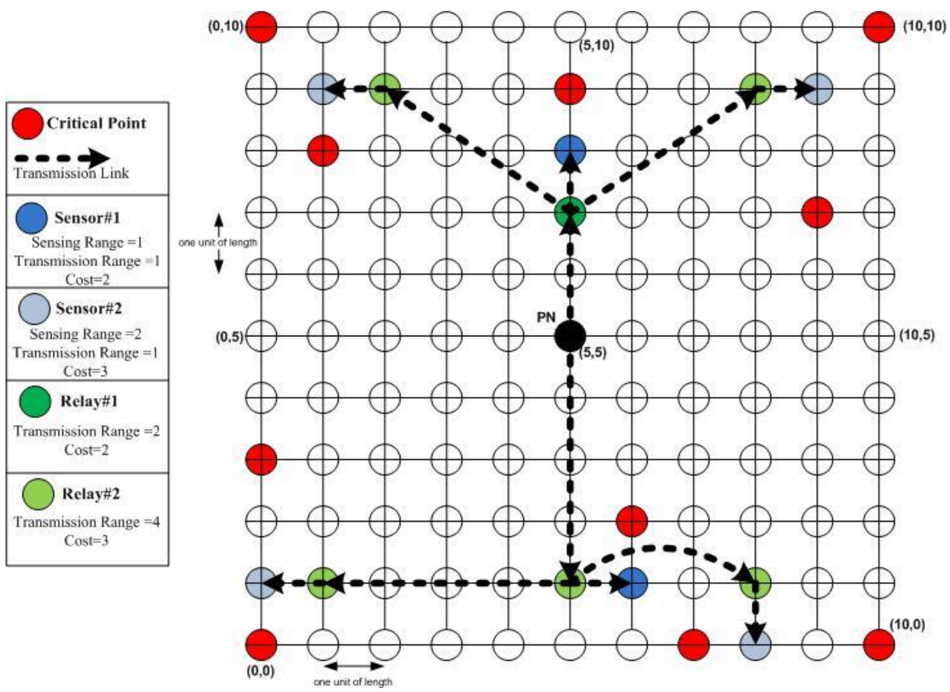

Figure 3-2: Bi-objective solution for Iteration 1

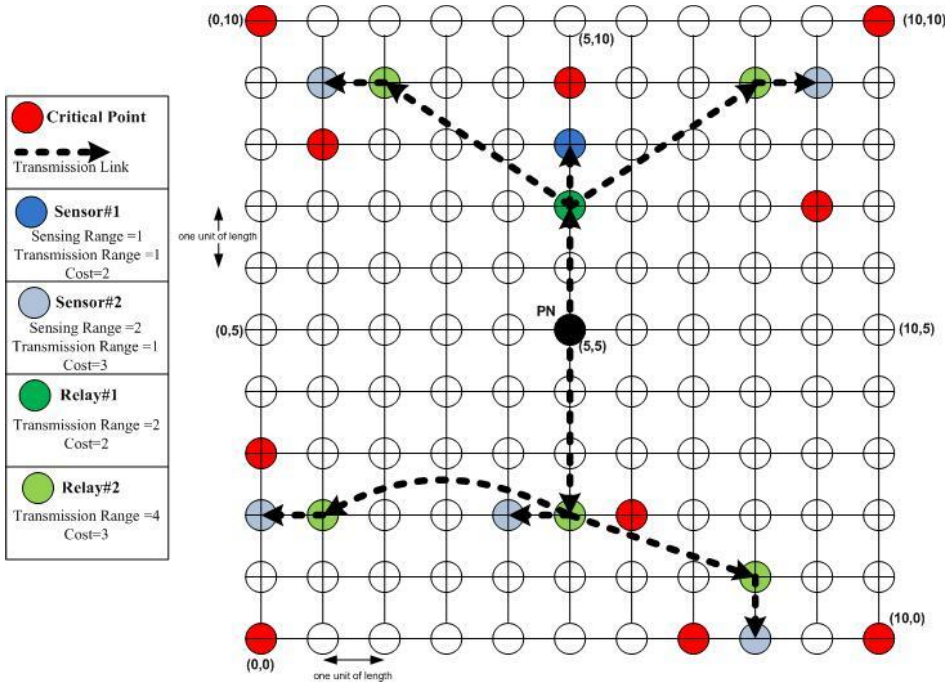

Figure 3-3: Bi-objective solution for Iteration 2

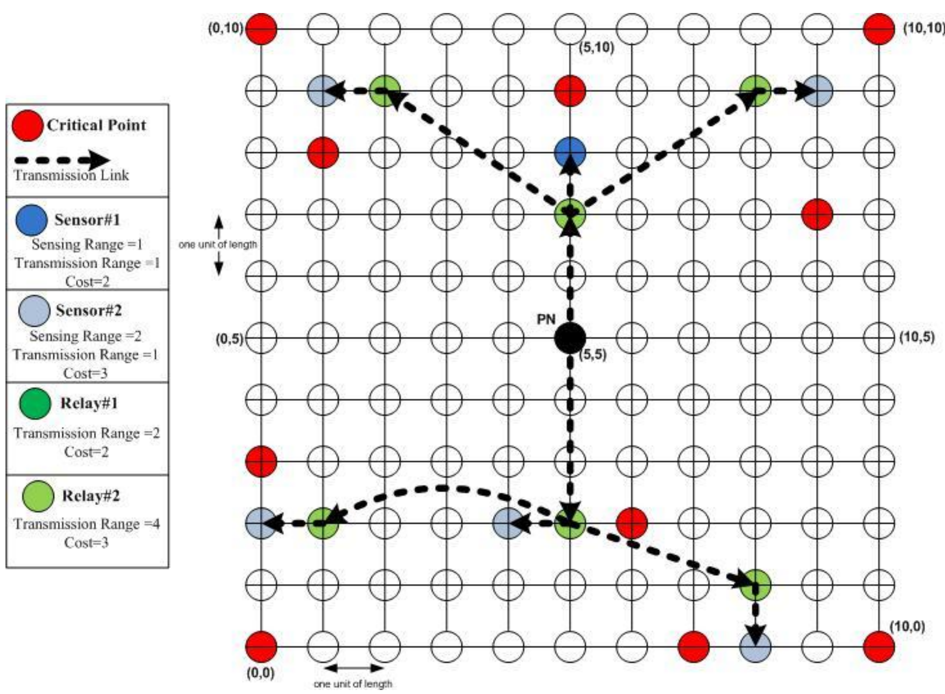

Figure 3-4: Bi-objective solution for Iteration 3

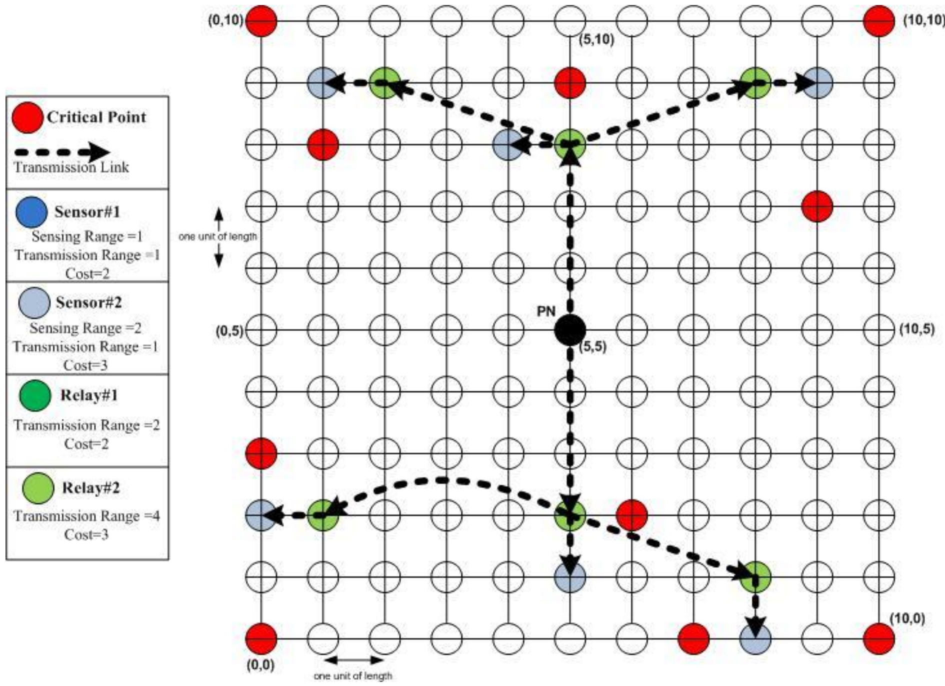

Figure 3-5: Bi-objective solution for Iteration 4

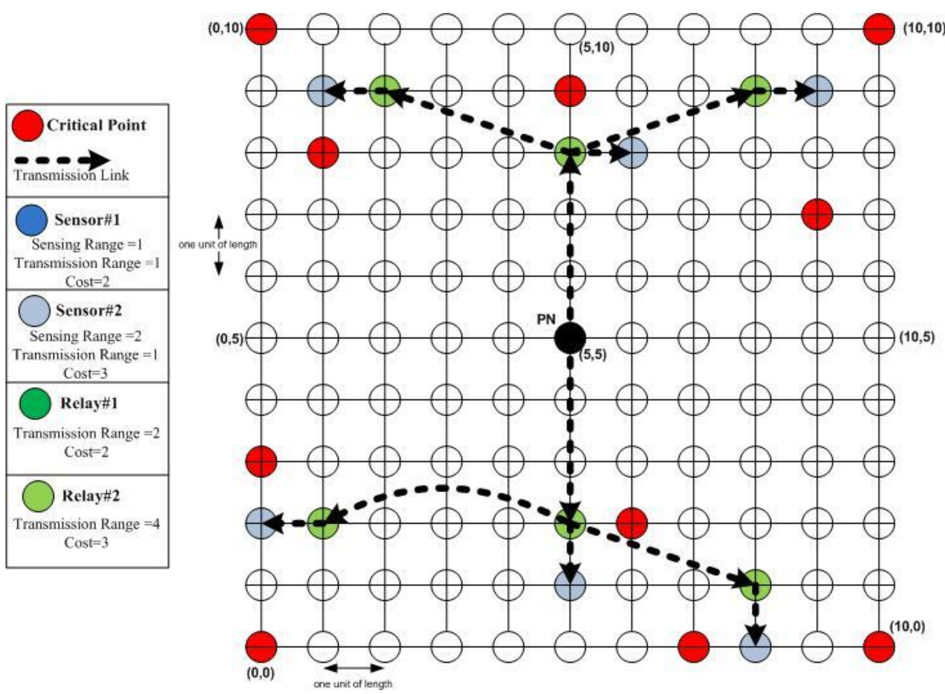

Figure 3-6: Bi-objective solution for Iteration 5

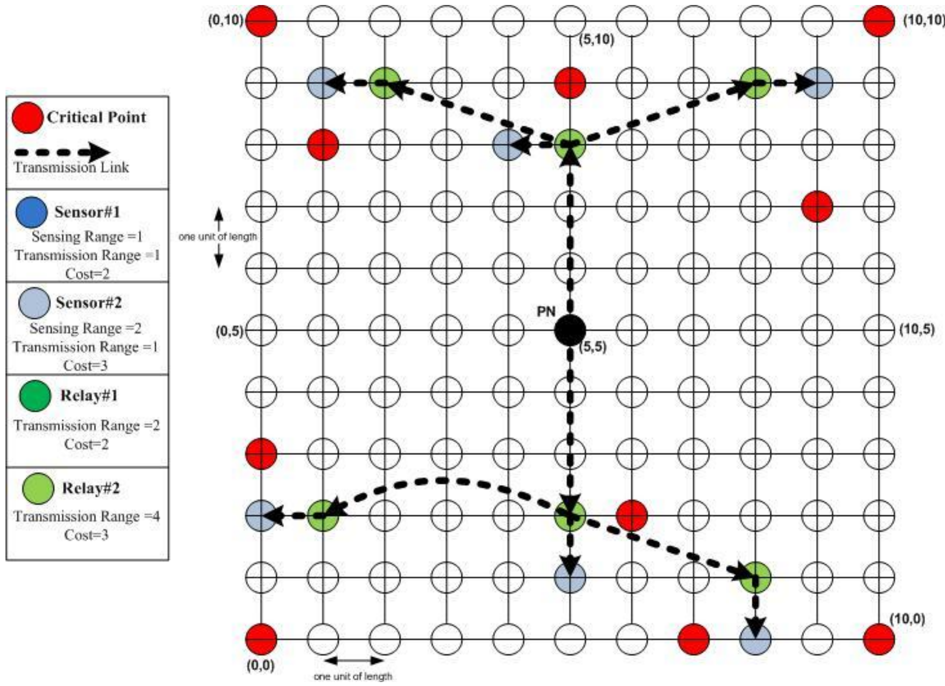

Figure 3-7: Bi-objective solution for Iteration 6

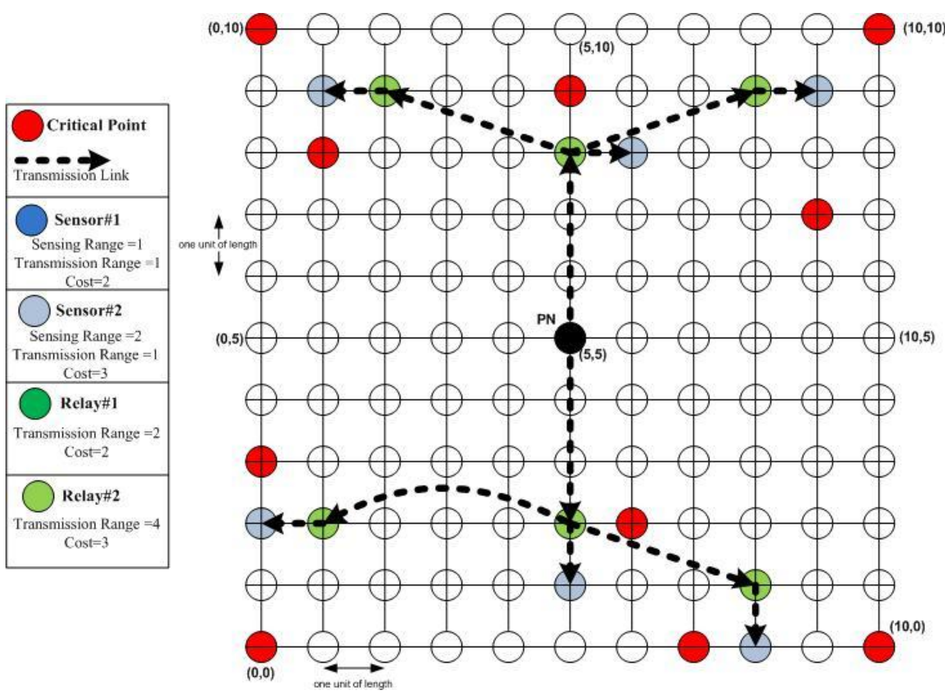

Figure 3-8: Bi-objective solution for Iteration 7

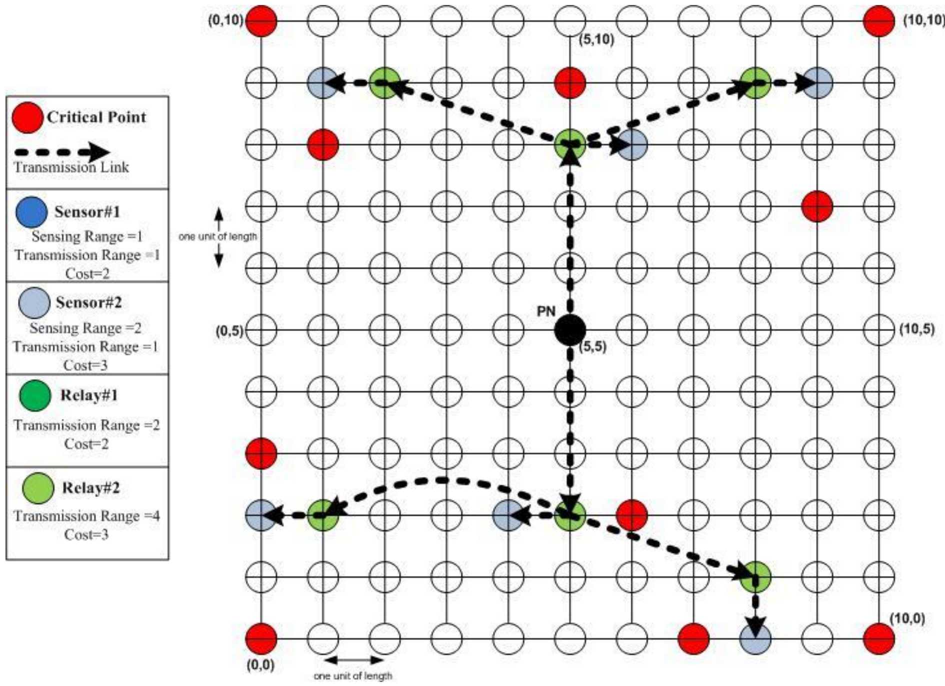

Figure 3-9: Bi-objective solution for Iteration 8

Chapter 4

SPACE PARTITIONING HEURISTIC METHOD

4.1 PREFACE

The previous examples illustrate an important aspect of sensor/relay deployment which is the mutual exclusiveness of most the deployed elements over the concerned space. In other words, the allocation of sensor/relay in position (i, j) can be done without the knowledge of the location of sensor/relay in position (k, l), for example. This means that the allocation of sensor/relay at (i, j) and (k, l) can be done in parallel. Therefore, to solve the sensor/relay problem, we do not need to program the whole problem as one ILP problem and solve at once. In contrast, we can partition the whole bounded space into sub-spaces of smaller dimensions and solve each one of them individually. Then, we compile all solutions together.

The resulting sub-spaces will be of smaller dimensions which will allow the integer linear program to solve them in relatively less amount of time than for solving the initial space directly. This method will obtain the solution much faster, depending on the decomposed sub-spaces dimensions, than the direct solving way. Despite the speedy solution obtained by the partitioning method, sub-optimal solutions are highly expected to be generated. For this reason, the partitioning method should be applied carefully so that optimality loss is minimized. Even though the integer linear program will not consider the whole space when solving in partitions, the method used for decomposition should minimize as much as possible the side effect of this heuristic.

In the following sections, we shall consider at the beginning few guidelines which help in executing this heuristic. Later, we introduce a method to enhance the optimality of the space-partitioning heuristic approach.

4.2 GUIDELINES TO USE SPACE PARTITIONING HEURISTIC METHOD

As mentioned previously, the space partitioning could yield sub-optimal solutions but in relatively faster computational time. For this reason, the space partitioning heuristic should be designed in a way such that sub-optimality is minimized to the possible lowest limit. In this section, some guidelines and suggestions are listed in order to be followed while implementing the space partitioning heuristic.

1. The partitioned space dimensions should not be less than the maximum transmissions range of all sensors and relays. This is basically, to allow for possible placement of any type of sensor or relay.

2. While partitioning the whole space, an effort should be made to include the processing node in each partitioned part. If the monitored area is initially large and it is impractical to include the processing node in each partitioned part, then a virtual processing node should be carefully placed on the same direction as the original processing node.

3. The method should start by partitioning the region with the highest concentration of critical points.

4.3 ALGORITHM B, APPLYING THE SPACE PARTITIONING HEURISTIC METHOD

4.3.1 INITIALIZATION PROCEDURE

Step 1: Partition the whole space into n smaller parts such that each part contains the processing node (In this stage, try to follow the previous guidelines in section 4.2 as much as possible).

Step 2: Apply the initialization procedure (Steps 1 – 4) as discussed in Algorithm A to each of the partitions. At the end of the iteration, state the bi-objective solution for each of the partitions in the following array:

$$OF^1 = \left\{\left(OF^1_{1,I}, OF^1_{1,III}\right), \left(OF^1_{2,I}, OF^1_{2,III}\right), \ldots, \left(OF^1_{n,I}, OF^1_{n,III}\right)\right\}$$

Where, $\left(OF^1_{i,I}, OF^1_{i,III}\right)$ refer to the bi-objective solution of the first iteration associated with the i^{th} partition.

4.3.2 ITERATIVE PROCEDURE

Step 3: For iteration i, if the following condition holds:

$$\left(\frac{{OF_{s,III}}^{i} - {OF_{s,III}}^{*}}{{OF_{s,III}}^{*}}\right) \times 100 \le \tau \quad , \quad \text{for all } s \text{ where } s = 1, \ldots, n$$

STOP the algorithm, otherwise go to Step 4.

Where τ is an acceptable percentage error or deviation of the current total energy value from the minimum total energy value.

Step 4: Set $i = i + 1$.

Step 5: For iteration i, set $OF_{s,I}{}^{i} = OF_{s,I}{}^{i-1} + k$ and solve Model III by substituting $OF_{s,I}{}^{i}$, for all s where: $s = 1,...,n$ on the right-hand side of constraint 3.13 to determine $OF_{s,III}{}^{i}$. Where k is a pre-selected increment. Now, combine the bi-objective solution of iteration i for each partition s where: $s = 1,...,n$ in the array:

$$OF^{i} = \left\{ \left(OF_{1,I}^{i}, OF_{1,III}^{i} \right), \left(OF_{2,I}^{i}, OF_{2,III}^{i} \right), ..., \left(OF_{n,I}^{i}, OF_{n,III}^{i} \right) \right\}.$$

Step 6: Let $\eta = \underset{1 \le s \le n}{argmax} \left\{ OF_{s,III}^{i-1} - OF_{s,III}^{i} \right\}$. Ties are broken arbitrarily. This step will basically determine the index number of the partition with the maximum reduction in the total energy and assign that index to the variable η.

The combined array of the final bi-objective solution for iteration i would be:

$$OF^{i} = \left\{ \left(OF_{1,I}^{i-1}, OF_{1,III}^{i-1} \right), \left(OF_{2,I}^{i-1}, OF_{2,III}^{i-1} \right), ..., \left(OF_{\eta,I}^{i}, OF_{\eta,III}^{i} \right), ..., \left(OF_{n,I}^{i-1}, OF_{n,III}^{i-1} \right) \right\} \text{ Where;}$$

$$OF_{I}^{i} = \sum_{\substack{1 \le s \le n \\ s \ne \eta}} OF_{s,I}^{i-1} + OF_{\eta,I}^{i} \qquad \text{and} \qquad OF_{II}^{i} = \sum_{\substack{1 \le s \le n \\ s \ne \eta}} OF_{s,II}^{i-1} + OF_{\eta,II}^{i}$$

Go to Step 3.

4.4 NUMERICAL EXAMPLE 2: FIRST SPACE PARTITIONING HEURISTIC – SCENARIO 1

This is an example on the use of the First Heuristic method to solve the problem presented in Example 1. The space partitioning heuristic method will be applied as discussed in Algorithm B to obtain the solution. The example is described below.

Given a 2D field of 10×10 unit length dimensions. The processing node is fixed at the location $(5,5)$. Ten critical points need to be covered with different levels of criticality. The critical points' location and criticality are given in Table 3-1. We assume the availability of two sensors and two relays in the market. The characteristics of each type of sensor and each type of relay were stated previously in Table 3-2 and Table 3-3 respectively.

In this example, the monitored space will be partitioned into four identical parts such that each part contains the processing node.

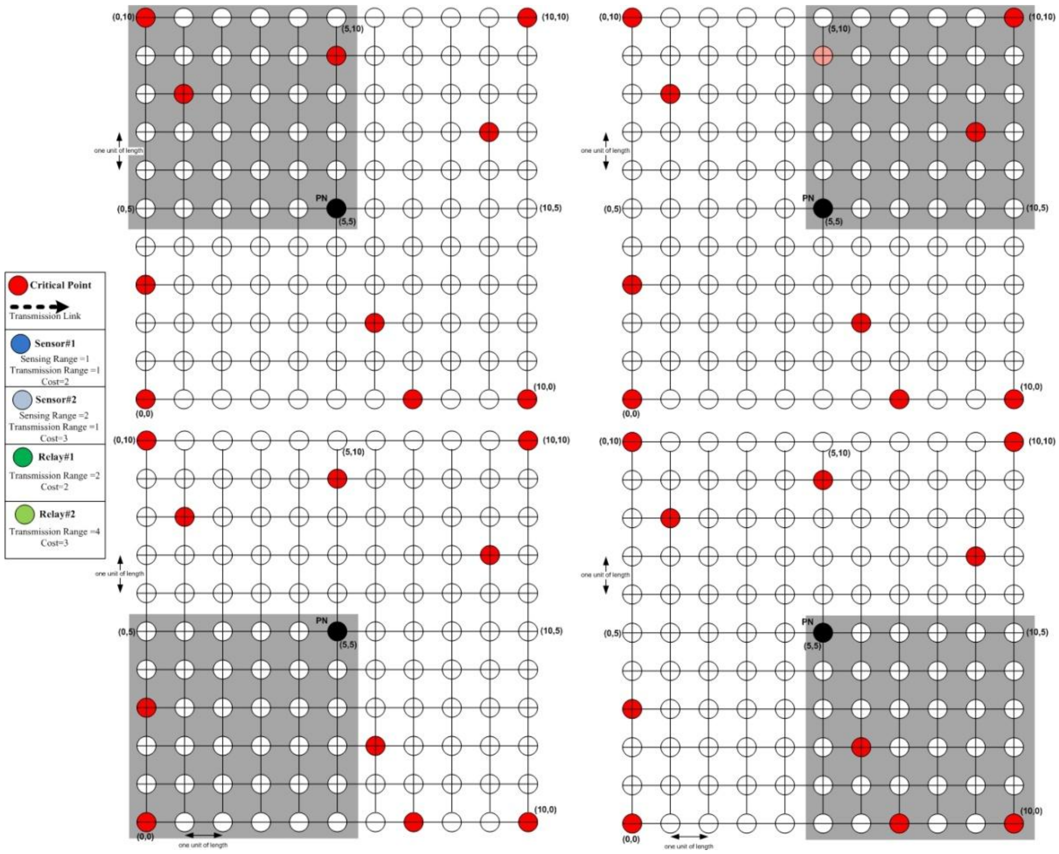

Figure 4-1: First heuristic partitions for Example 2

Note: In this example, the critical point at $(5,9)$ is shared by partitions 3 and 4 simultaneously. The ILP model needs to detect each critical point once. Thus, this critical point can be considered in one partition only. When the other partition is solved, the critical point is not inserted in the model since it has already been involved in another partition.

In this example, we consider the critical point at $(5,9)$ as a part of partition 3. In the next example, i.e. Example 3, the same critical point will be considered as a part of partition 4 in an attempt to observe the effect of the two scenarios on the solution.

The previous figure, i.e. Figure 4-1, demonstrates visually the different partitions. The pink-colored critical point in partition 4 indicates that this critical point has already been considered and will not be included again in this partition.

	First Space Partitioning Heuristic Solution		**Iteration Solution**	
Iteration	OF_I **= Network Cost ($)**	OF_{II} **= Total Energy Consumption (mJ)**	OF_I **= Network Cost ($)**	OF_{II} **= Total Energy Consumption (mJ)**
First Iteration, OF^1	$OF^*_{1,I} = 8$	0.24128	8	0.24128
	$OF^*_{2,I} = 10$	0.32136	10	0.32136
	$OF^*_{3,I} = 10$	0.32152	10	0.32152
	$OF^*_{4,I} = 8$	0.24128	8	0.24128
Iteration Solution			**36**	**1.12544**
Second Iteration, OF^2	9	0.24112	9	0.24112
	11	0.32120	10	0.32136
	11	0.32136	10	0.32152
	9	0.24112	8	0.24128
Iteration Solution			**37**	**1.12528**
Third Iteration, OF^3	10	0.24112	9	0.24112
	11	0.32120	11	0.32120
	11	0.32136	10	0.32152
	9	0.24112	8	0.24128

Iteration Solution			**38**	**1.12512**
Fourth Iteration, OF^4	11	0.24112	9	0.24112
	12	0.32120	11	0.32120
	11	0.32136	11	0.32136
	9	0.24112	8	0.24128
Iteration Solution			**39**	**1.12496**
Fifth Iteration, OF^5	12	0.24112	9	0.24112
	13	0.32120	11	0.32120
	12	0.32120	12	0.32120
	9	0.24112	8	0.24128
Iteration Solution			**40**	**1.12480**
Sixth Iteration, OF^6	----------	-----------	9	0.24112
	14	0.32120	11	0.32120
	13	0.32120	12	0.32120
	9	0.24112	9	0.24112
Iteration Solution			**41**	**1.12464**
Seventh Iteration, OF^7	----------	-----------	9	0.24112
	----------	-----------	11	0.32120
	14	0.32120	12	0.32120
	10	0.24112	9	0.24112
Iteration Solution			**41**	**1.12464**
Eighth Iteration, OF^8	----------	-----------	9	0.24112
	----------	-----------	11	0.32120
	15	0.32120	12	0.32120
	11	0.24112	9	0.24112
Iteration Solution			**41**	**1.12464**
Ninth Iteration, OF^9	----------	-----------	9	0.24112
	----------	-----------	11	0.32120
	----------	-----------	12	0.32120
	12	0.24112	9	0.24112
Iteration Solution			**41**	**1.12464**

Final Solution	----------	-----------	9	0.24112
	----------	-----------	11	0.32120
	----------	-----------	12	0.32120
	----------	-----------	9	0.24112
Iteration Solution			**41**	**1.12464**

Table 4-1: First heuristic method's solution

We terminate the heuristic in any partition when adding the maximum cost (through sensors and relays) does not reduce the total network power consumption. In our example, the maximum cost (among all sensors and relays) is 3. So, if increasing the cost upper limit in any partition does not reduce the power consumption in that particular partition, the partition is terminated and no more iteration is done.

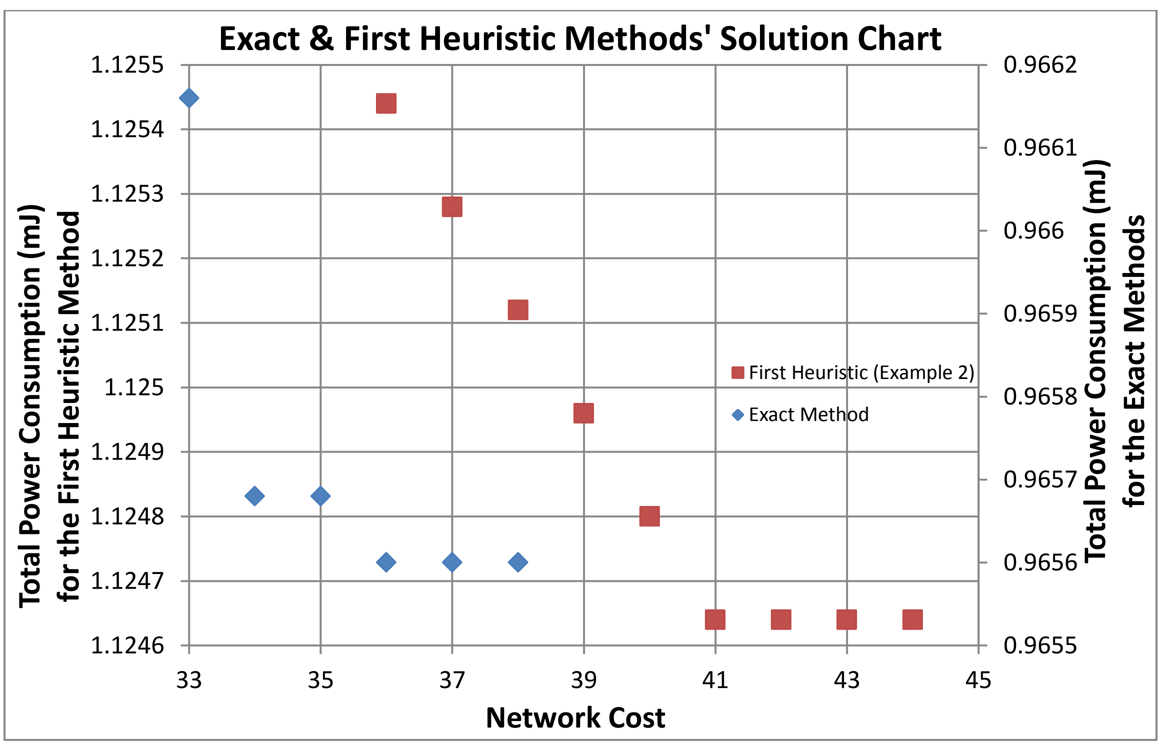

Figure 4-2: Exact and first heuristic methods' performance

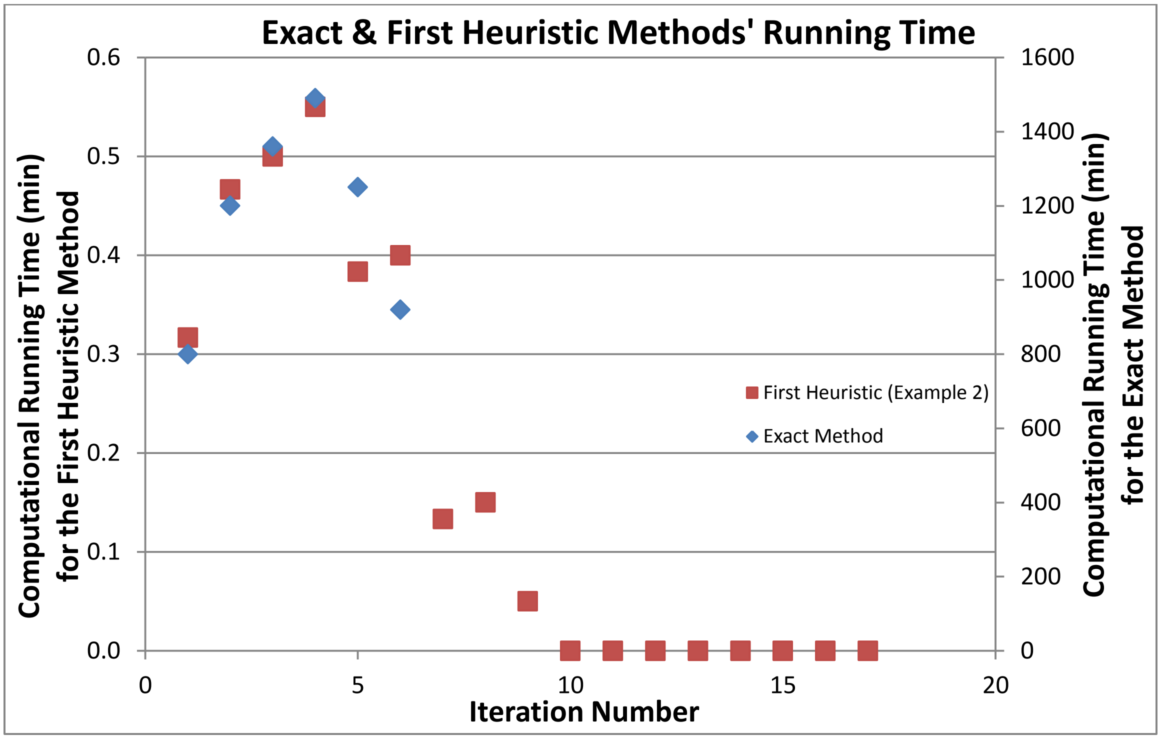

Figure 4-3: Exact and first heuristic methods' running time

4.5 NUMERICAL EXAMPLE 3: FIRST SPACE PARTITIONING HEURISTIC – SCENARIO 2

The problem discussed in Example 1 has a critical point that is common to partition 3 and partition 4. The space partitioning heuristic states that such a critical point is considered as belonging to only one of the two partitions. In this example, we investigate the effect of considering this common critical point with partition 4 as opposed to considering it with partition 3 as done in Example 2. Then, the solutions obtained from both scenarios, i.e. including the common critical point with partition 3 and then with partition 4, will be compared.

As described earlier, the problem space is a 2D field of 10×10 unit length dimensions. The processing node is fixed at the location $(5,5)$. Ten critical points need to be covered with different levels of criticality. The critical points' location and criticality are given in Table 3-1. We assume the availability of two sensors and two relays in the market. The characteristics of each type of sensor and each type of relay were stated previously in Table 3-2 and Table 3-3 respectively.

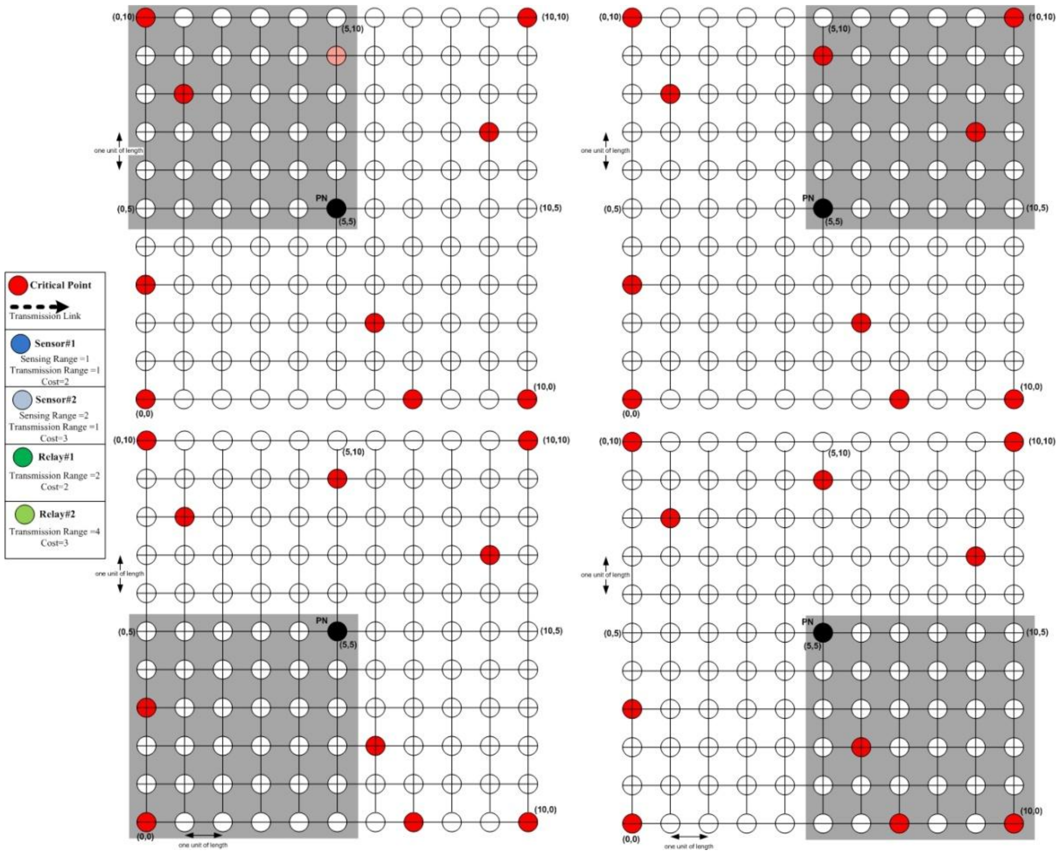

Figure 4-4: First heuristic partitions for Example 3

Note: In this example, the same problem in Example 2 is solved again but with the critical point at $(5,9)$ being considered as a part of partition 3. The partitions are demonstrated visually in the previous figure, i.e. Figure 4-4. The pink-colored critical point in partition 3 indicates that this critical point has already been considered and will not be included again in this partition.

	First Space Partitioning Heuristic Solution				**Iteration Solution**	
	Common critical point belongs to Q3 (Example 2)		**Common critical point belongs to Q4**			
Iteration	OF_I **($)**	OF_{II} **(mJ)**	OF_I **($)**	OF_{II} **(mJ)**	OF_I **($)**	OF_{II} **(mJ)**
First Iteration, OF^1	$OF^*_{1,I} = 8$	0.24128	$OF^1_{1,I} = 8$	0.24128	8	0.24128
	$OF^*_{2,I} = 10$	0.32136	$OF^1_{2,I} = 10$	0.32136	10	0.32136
	$OF^*_{3,I} = 10$	0.32152	$OF^1_{3,I} = 8$	0.24128	8	0.24128
	$OF^*_{4,I} = 8$	0.24128	$OF^1_{4,I} = 10$	0.32152	10	0.32152
Iteration Solution					**36**	**1.12544**
Second Iteration, OF^2	9	0.24112	9	0.24112	9	0.24112
	11	0.32120	11	0.32120	10	0.32136
	11	0.32136	9	0.24112	8	0.24128
	9	0.24112	11	0. 32136	10	0.32152
Iteration Solution					**37**	**1.12528**
Third Iteration, OF^3	10	0.24112	10	0.24112	9	0.24112
	11	0.32120	11	0.32120	11	0.32120
	11	0.32136	9	0.24112	8	0.24128
	9	0.24112	11	0. 32136	10	0.32152
Iteration Solution					**38**	**1.12512**
Fourth Iteration, OF^4	11	0.24112	11	0.24112	9	0.24112
	12	0.32120	12	0.32120	11	0.32120
	11	0.32136	9	0.24112	9	0.24112
	9	0.24112	11	0. 32136	10	0.32152
Iteration Solution					**39**	**1.12496**
Fifth Iteration, OF^5	12	0.24112	12	0.24112	9	0.24112
	13	0.32120	13	0.32120	11	0.32120
	12	0.32120	10	0.24112	9	0.24112
	9	0.24112	11	0. 32136	11	0.32136
Iteration Solution					**40**	**1.12480**

Sixth Iteration, OF^6	----------	-----------	----------	-----------	9	0.24112
	14	0.32120	14	0.32120	11	0.32120
	13	0.32120	11	0.24112	9	0.24112
	9	0.24112	12	0. 32120	12	0.32120
Iteration Solution					**41**	**1.12464**
Seventh Iteration, OF^7	----------	-----------	----------	-----------	9	0.24112
	----------	-----------	----------	-----------	11	0.32120
	14	0.32120	12	0.24112	9	0.24112
	10	0.24112	13	0. 32120	12	0.32120
Iteration Solution					**41**	**1.12464**
Eighth Iteration, OF^8	----------	-----------	----------	-----------	9	0.24112
	----------	-----------	----------	-----------	11	0.32120
	15	0.32120	----------	-----------	9	0.24112
	11	0.24112	14	0. 32120	12	0.32120
Iteration Solution					**41**	**1.12464**
Ninth Iteration, OF^9	----------	-----------	----------	-----------	9	0.24112
	----------	-----------	----------	-----------	11	0.32120
	----------	-----------	----------	-----------	9	0.24112
	12	0.24112	15	0. 32120	12	0.32120
Iteration Solution					**41**	**1.12464**
Final Solution	----------	-----------	----------	-----------	9	0.24112
	----------	-----------	----------	-----------	11	0.32120
	----------	-----------	----------	-----------	9	0.24112
	----------	-----------	----------	-----------	12	0.32120
Iteration Solution					**41**	**1.12464**

Table 4-2: First heuristic method's solution for two scenarios

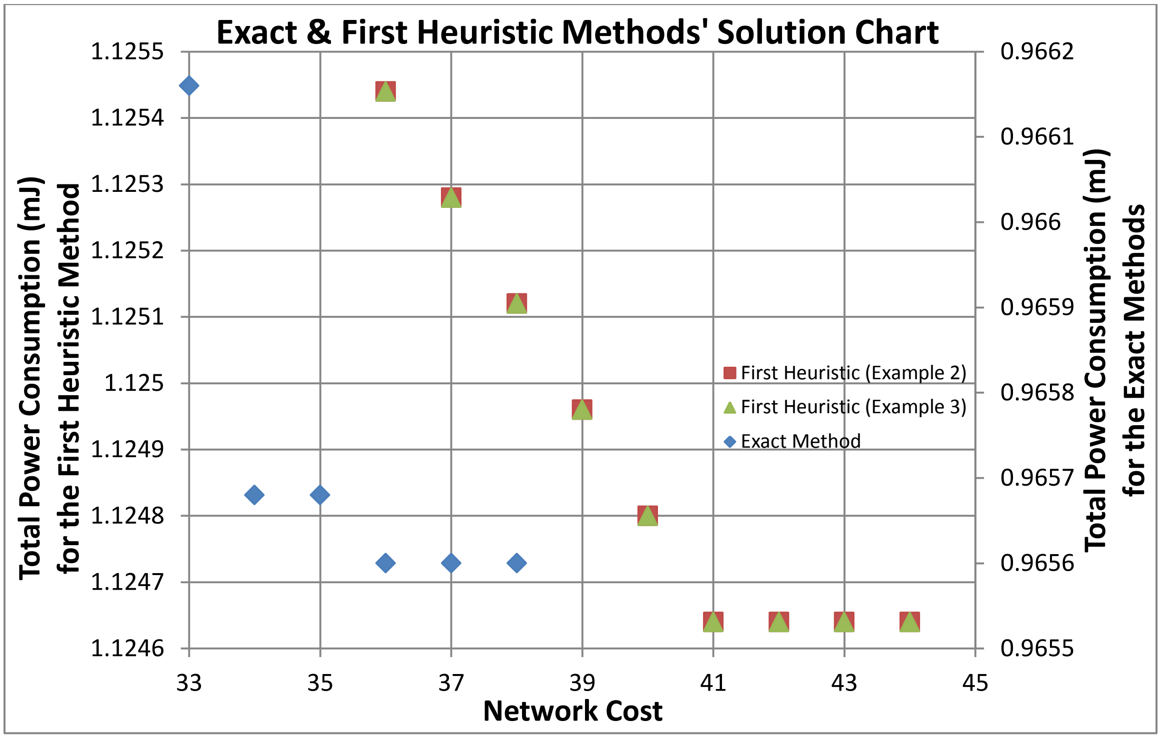

Figure 4-5: Exact and the two-scenarios first heuristic methods' performance

The previous graph shows the solutions obtained by the exact and heuristic methods to solve the problem described in Example 1. The right vertical axis corresponds to the energy consumption obtained by the exact method. Likewise, the left vertical axis corresponds to the energy consumption obtained by the heuristic method. For the heuristic method, two scenarios for were considered. In each scenario the common critical point between partitions 3 and 4 was considered in one of them. It is obvious that in this example no improvement was obtained by considering the two scenarios and their solutions are equivalent. Differently, the solution obtained by the exact method is better, i.e. yields less energy consumption for the same total cost compared to the heuristic method, than the heuristic method's solution. This result motivates us to enhance the space partitioning heuristic method to obtain better solutions. This modified method will be presented in the next example.

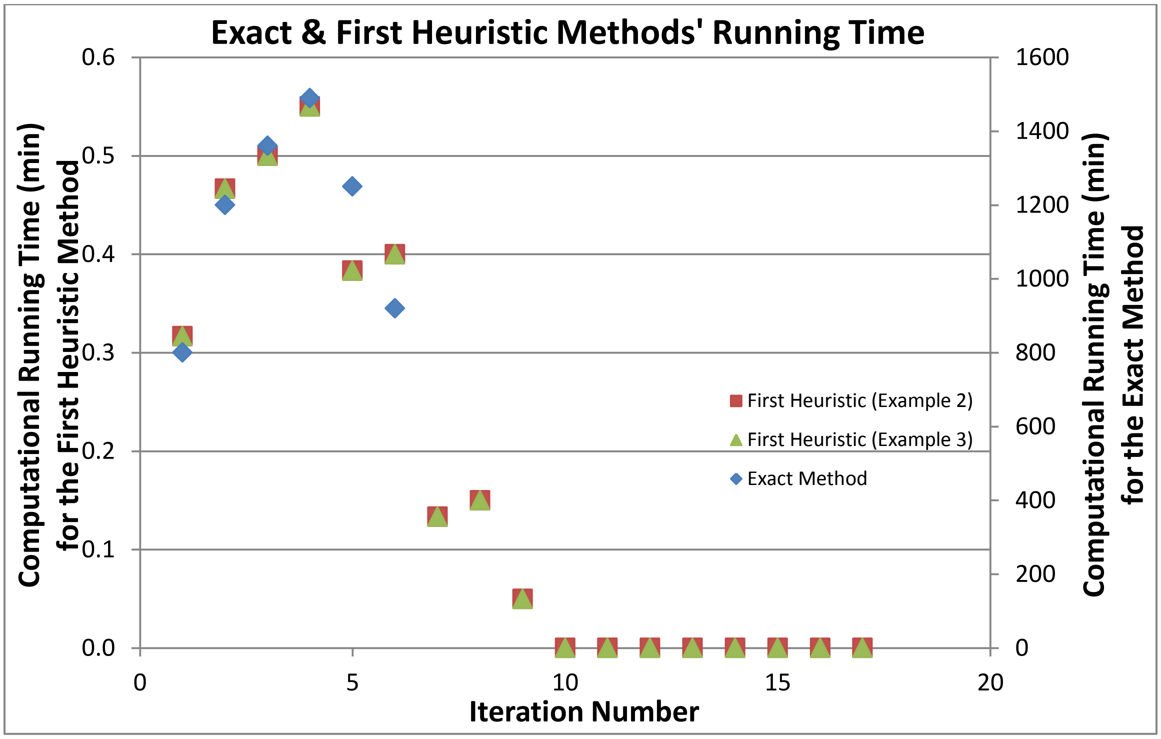

Figure 4-6: Exact and the two-scenarios first heuristic methods' running time

From the previous graph, the time consumed by the exact method (shown in the right axis) is much larger than the time consumed by the heuristic method (shown in the left axis). Even though no improvement is obtained by using either way over the other, the user should put an effort to consider such possible configurations as this might result in substantial improvement in the ultimate solution.

4.6 NUMERICAL EXAMPLE 4: SECOND SPACE PARTITIONING HEURISTIC

This example is intended to solve the problem discussed in Example 1 using the Second Heuristic method (a modified procedure of the initial Heuristic method). The method goes as follows: first the space is partitioned and each partition is solved separately as presented in Algorithm B. Then, the whole solution is combined. Finally, to form the final solution for the iteration, the solution in the margin area that separates the partitions is deleted and the whole space is solved again as one space. The Second Heuristic method is expected to provide better solutions than the initial First Heuristic method. The reason is that by deleting the solution in the common margin area and solving the whole space again, the integer program gets the opportunity to realize the complete problem, obtain more information and thus produce better results. The next example will illustrate the advantage of using the Second Heuristic method.

Given a 2D field of 10×10 unit length dimensions. The processing node is fixed at the location $(5,5)$. Ten critical points need to be covered with different levels of criticality. The critical points' location and criticality are given in Table 3-1. We assume the availability of two sensors and two relays in the market. The characteristics of each type of sensor and each type of relay were stated previously in Table 3-2 and Table 3-3 respectively.

The next graph shows the margin area selected in solving the problem for this example. Then, a table showing the solution for each iteration is presented.

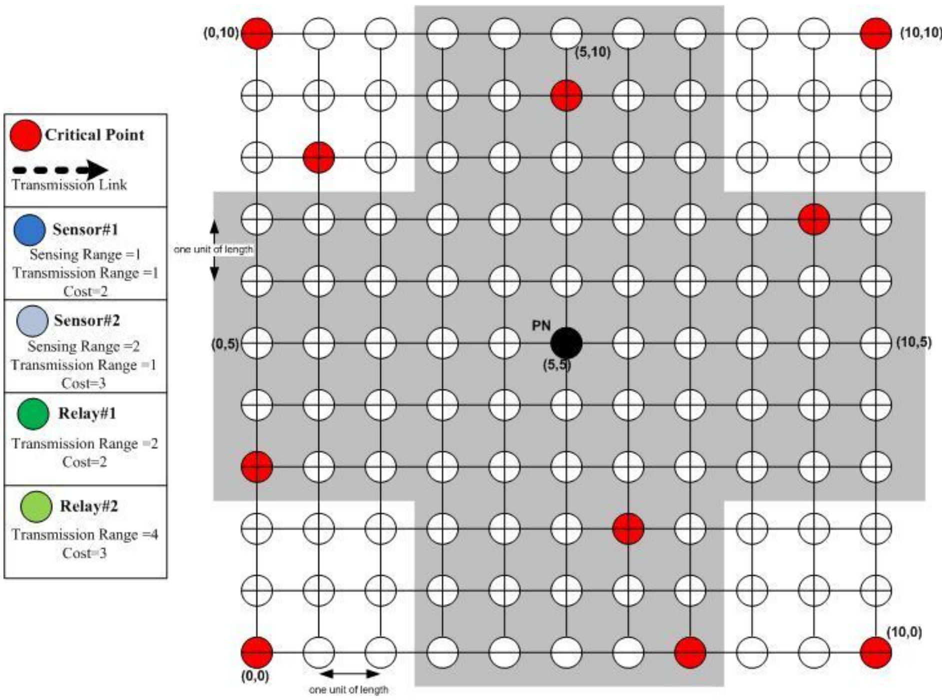

Figure 4-7: The margin area for Example 4

The solution of Example 4 is listed in the following table:

	Second Space Partitioning Heuristic Solution		**Iteration Solution**	
Iteration	OF_I = **Network Cost ($)**	OF_{II} = **Total Energy Consumption (mJ)**	OF_I = **Network Cost ($)**	OF_{II} = **Total Energy Consumption (mJ)**
First Iteration, OF^1	$OF^*_{1,I}$: 8	0.24128	8	
	$OF^*_{2,I}$: 10	0.32136	10	
	$OF^*_{3,I}$: 8	0.24128	8	
	$OF^*_{4,I}$: 10	0.32152	10	
Iteration Solution			**36**	**1.04496**
Second Iteration, OF^2	9	0.24112	9	
	11	0.32120	10	
	9	0.24112	8	

	11	0. 32136	10	
Iteration Solution			**37**	**0.96568**
Third Iteration, OF^3	10	0.24112	9	
	11	0.32120	11	
	9	0.24112	8	
	11	0. 32136	10	
Iteration Solution			**38**	**0.96560**
Fourth Iteration, OF^4	11	0.24112	9	
	12	0.32120	11	
	9	0.24112	9	
	11	0. 32136	10	
Iteration Solution			**39**	**0.96560**
Fifth Iteration, OF^5	12	0.24112	9	
	13	0.32120	11	
	10	0.24112	9	
	11	0. 32136	11	
Iteration Solution			**40**	**0.96560**
Sixth Iteration, OF^6	----------	-----------	9	
	14	0.32120	11	
	11	0.24112	9	
	12	0. 32120	12	
Iteration Solution			**41**	**0.96560**
Seventh Iteration, OF^7	----------	-----------	9	
	----------	-----------	11	
	12	0.24112	9	
	13	0. 32120	12	
Iteration Solution			**41**	**0.96560**
Eighth Iteration, OF^8	----------	-----------	9	
	----------	-----------	11	
	----------	-----------	9	
	14	0. 32120	12	

Iteration Solution			**41**	**0.96560**
Ninth Iteration, OF^9	----------	-----------	9	
	----------	-----------	11	
	----------	-----------	9	
	15	0. 32120	12	
Iteration Solution			**41**	**0.96560**
Final Solution	-----------	----------	9	
	-----------	----------	11	
	-----------	----------	9	
	-----------	----------	12	
Iteration Solution				**0.96560**

Table 4-3: Second heuristic method's solution

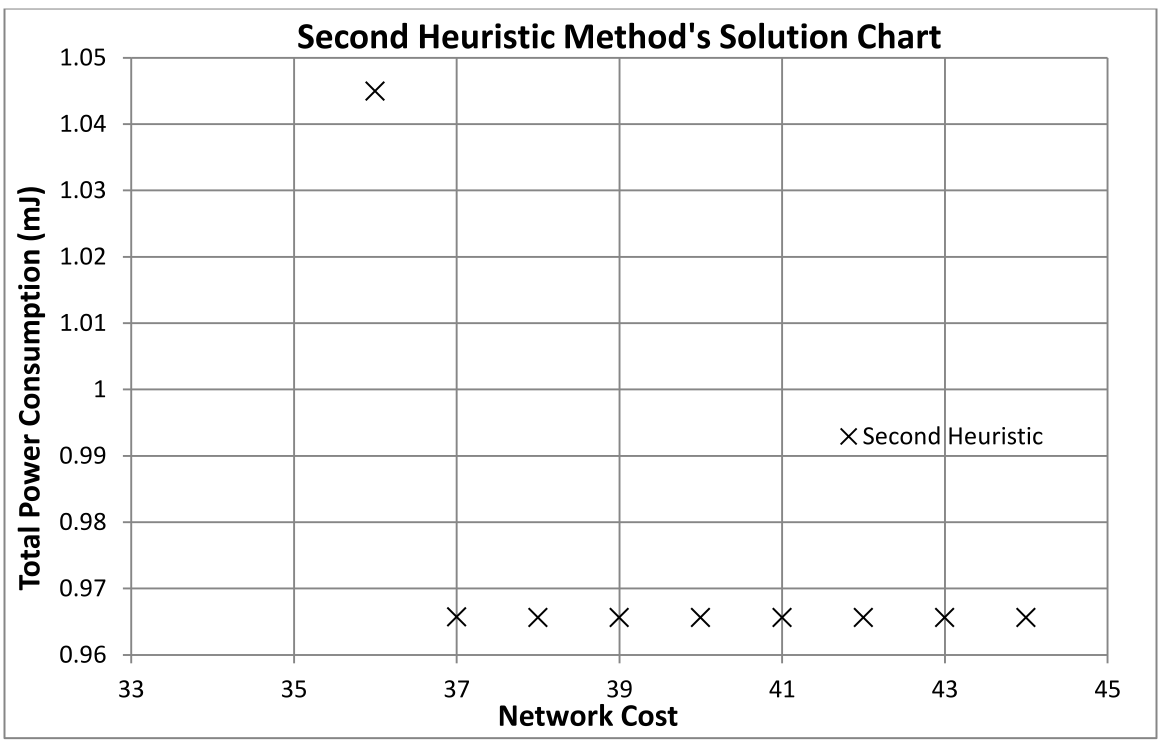

Figure 4-8: Second heuristic method's performance

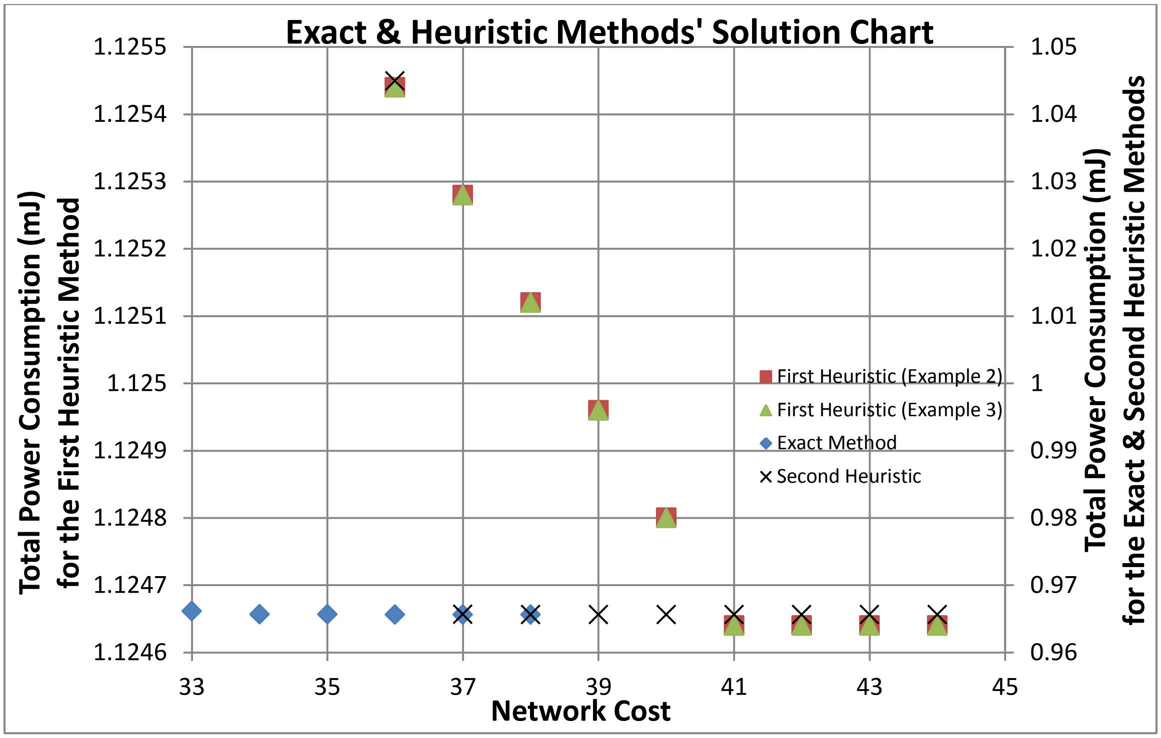

Figure 4-9: Exact and heuristic methods' performance

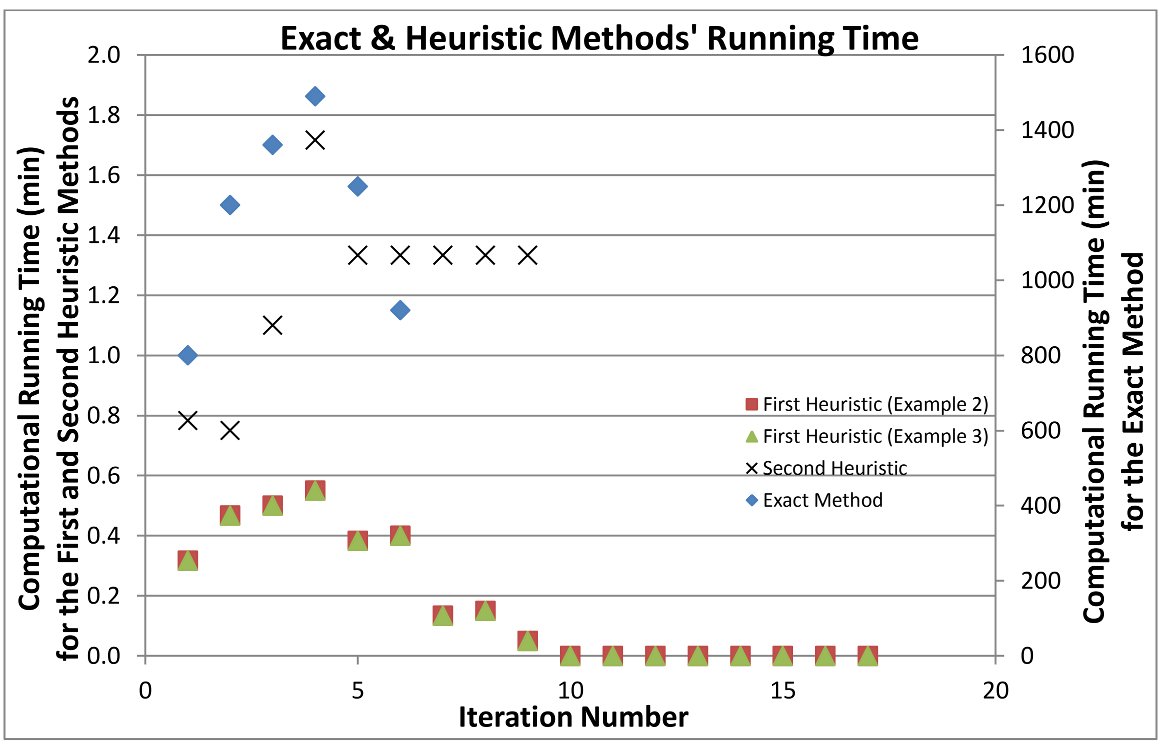

Figure 4-10: Exact and heuristic methods' running time

The previous two charts present the solutions obtained by the exact, first heuristic and second heuristic methods. It also shows the running time consumed by each of the methods. It can be concluded that the solution obtained by the second heuristic method will not be worse than the solution obtained by the first heuristic method in any case. In this example, the solution of the second heuristic method (shown in the right vertical axis) dominates the solution of the first heuristic method (shown in the left vertical axis). The computational running time for the second heuristic method is much less than the exact method's but a little more than the first heuristic method's. The unique property of the second heuristic method is that it compromises between the powerfulness of the exact method, i.e. obtaining the optimal solution, and the first heuristic method, i.e. obtaining a fast solution. The larger the common margin area is, the more time it takes it solve the whole area problem and the closer the solution the exact method's solution. On the other hand, the smaller the common margin area is, the less time it takes to solve the whole area problem and the closer the solution to the first heuristic method's solution.

4.7 NUMERICAL EXAMPLE 5: FIRST HEURISTIC FOR LARGE SPACE

In this example, the idea of dealing with large spaces will be illustrated. We will attempt to solve a 2D field of 20×20 unit length dimensions using the First Heuristic method. The processing node is fixed at the location $(10,10)$. We assume the availability of two sensors and two relays as presented previously in Table 3-2 and Table 3-3. The location and criticality of each critical point are listed in the following table:

Critical Point	1	2	3	4	5	6	7	8	9	10
Location	(8,0)	(5,1)	(9,1)	(16,2)	(18,3)	(2,4)	(5,4)	(19,5)	(0,6)	(6,7)
Criticality	1	1	1	1	1	1	1	1	1	1
Critical Point	11	12	13	14	15	16	17	18	19	20
Location	(3,7)	(8,7)	(14,7)	(2,8)	(15,8)	(14,9)	(18,9)	(19,10)	(12,11)	(20,11)
Criticality	1	1	1	1	1	1	1	1	1	1
Critical Point	21	22	23	24	25	26	27	28	29	30
Location	(1,12)	(6,12)	(7,12)	(13,13)	(16,13)	(2,15)	(6,15)	(10,15)	(1,16)	(18,17)
Criticality	1	1	1	1	1	1	1	1	1	1
Critical Point	31	32	33	34	35					
Location	(9,19)	(16,19)	(17,19)	(0,20)	(15,20)					
Criticality	1	1	1	1	1					

Table 4-4: Critical points' locations and criticalities for Example 4

The 2D space with all the critical points and the Processing Node are shown clearly in the following graph:

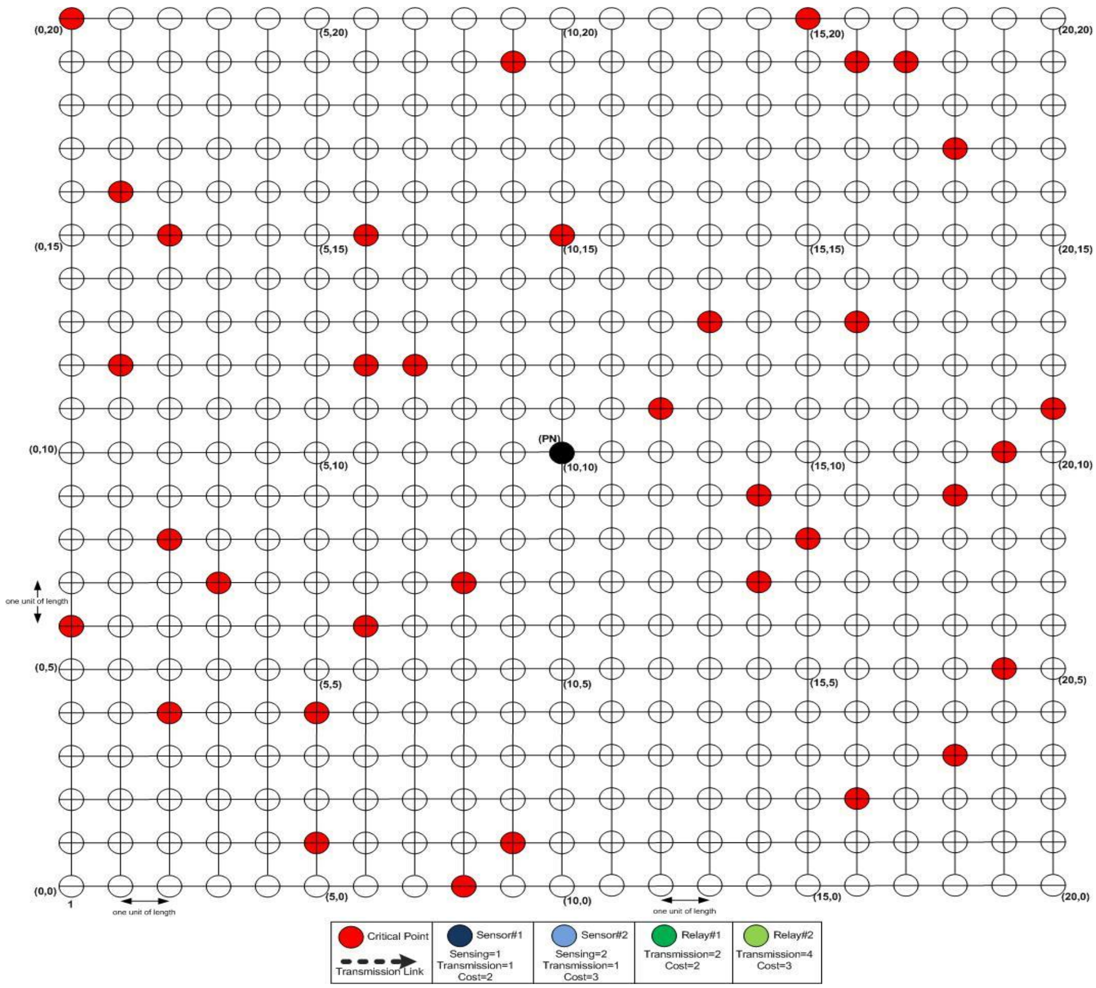

Figure 4-11: Critical points locations for Example 4

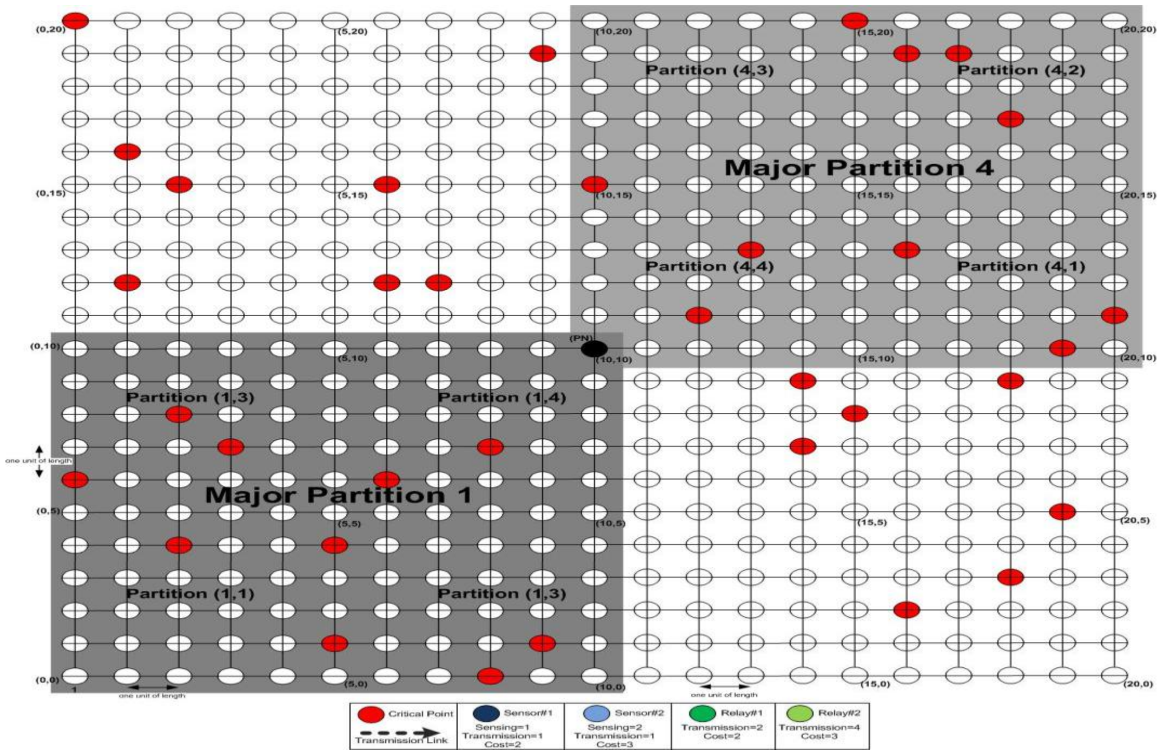

Figure 4-12: Partitions 1 and 4 in Example 4

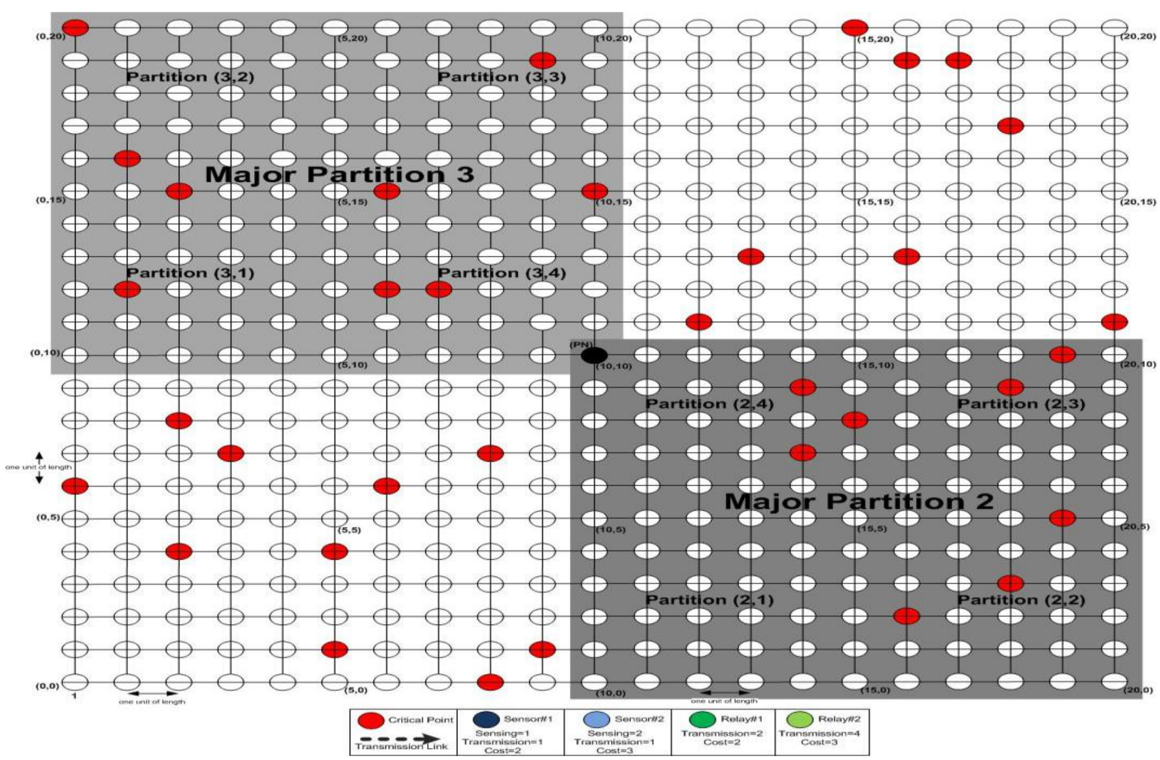

Figure 4-13: Partitions 2 and 3 in Example 4

The space is partitioned into four major partitions. Each major partition is again divided into another four partitions. The total number of partitions is sixteen. To start the solution, each of the sixteen partitions is solved independently and the solutions are combined to form the initial whole solution. Then the major four partitions are solved in each subsequent iteration and the partition with the maximum reduction in power is reserved while the others take the previous iteration solution as discussed previously in Algorithm B.

The following table lists the solutions obtained in each iteration using the First Heuristic Method:

	First Space Partitioning Heuristic Solution		**Iteration Solution**	
Iteration	OF_I **= Network Cost ($)**	OF_{II} **= Total Energy Consumption (mJ)**	OF_I **= Network Cost ($)**	OF_{II} **= Total Energy Consumption (mJ)**
First Iteration, OF^1	$OF^*_{(1,1),I} = 9$	0.32048	9	0.32048
	$OF^*_{(1,2),I} = 5$	0.16112	8	0.16112
	$OF^*_{(1,3),I} = 8$	0.24088	8	0.24088
	$OF^*_{1,I} = 31$	1.04440	33	1.04440
	OF^1_1		**33**	**1.04440**
	$OF^*_{(2,1),I} =$	---------	---------	---------
	$OF^*_{(2,2),I} = 9$	0.24056	9	0.24056
	$OF^*_{(2,3),I} = 6$	0.24048	6	0.24048
	$OF^*_{2,I} = 26$	0.88384	27	0.88384
	OF^1_2		**27**	**0.88384**

	$OF^{*}_{(3,1),I} = 6$	0.16112	6	0.16112
	$OF^{*}_{(3,2),I} = 10$	0.32136	10	0.32136
	$OF^{*}_{(3,3),I} = 7$	0.24048	7	0.24048
	$OF^{*}_{3,I} = 39$	1.12472	39	1.12472
	OF^{1}_{3}		**39**	**1.12472**
	$OF^{*}_{(4,1),I} = 8$	0.32064	8	0.32064
	$OF^{*}_{(4,2),I} = 10$	0.32056	10	0.32056
	$OF^{*}_{(4,3),I}$	---------	---------	---------
	$OF^{*}_{4,I} = 29$	0.96360	29	0.96360
	OF^{1}_{4}		**29**	**0.96360**
Iteration Solution			**128**	**4.01656**
Second Iteration, OF^{2}	$OF^{2}_{(1,1),I} = 10$	0.80416		
	$OF^{2}_{(1,2),I} = 6$	1.04408		
	$OF^{2}_{(1,3),I} = 9$	0.96424		
	$OF^{2}_{1,I} = 34$	1.04432		
	OF^{2}_{1}		**34**	**0.80416**
	$OF^{2}_{(2,1),I} =$	---------		
	$OF^{2}_{(2,2),I} = 10$	0.88352		
	$OF^{2}_{(2,3),I} = 7$	0.88360		
	$OF^{2}_{2,I} = 28$	0.88336		
	OF^{2}_{2}		**27**	**0.88384**
	$OF^{2}_{(3,1),I} = 7$	1.12416		
	$OF^{2}_{(3,2),I} = 11$	1.12448		
	$OF^{2}_{(3,3),I} = 8$	1.12448		

	$OF^{2}_{3,I}$ =40	1.12472		
	OF^{2}_{3}		**39**	**1.12472**
	$OF^{2}_{(4,1),I}$ =9	0.88352		
	$OF^{2}_{(4,2),I}$ =11	0.9636		
	$OF^{2}_{(4,3),I}$	---------		
	$OF^{2}_{4,I}$ =30	0.9632		
	OF^{2}_{4}		**29**	**0.96360**
Iteration Solution			**129**	**3.77632**
Third Iteration, OF^{3}	$OF^{3}_{(1,1),I}$ =11	0.80416		
	$OF^{3}_{(1,2),I}$ =6	0.80392		
	$OF^{3}_{(1,3),I}$ =9	0.72400		
	$OF^{3}_{1,I}$ =35	0.80400		
	OF^{3}_{1}		**35**	**0.72400**
	$OF^{3}_{(2,1),I}$ =	---------		
	$OF^{3}_{(2,2),I}$ =10	0.88352		
	$OF^{3}_{(2,3),I}$ =7	0.88360		
	$OF^{3}_{2,I}$ =28	0.88336		
	OF^{3}_{2}		**27**	**0.88384**
	$OF^{3}_{(3,1),I}$ =7	1.12416		
	$OF^{3}_{(3,2),I}$ =11	1.12448		
	$OF^{3}_{(3,3),I}$ =8	1.12448		
	$OF^{3}_{3,I}$ =40	1.12472		
	OF^{3}_{3}		**39**	**1.12472**
	$OF^{3}_{(4,1),I}$ =9	0.88352		

	$OF^{3}_{(4,2),I}$ =11	0.96360		
	$OF^{3}_{(4,3),I}$	---------		
	$OF^{3}_{4,I}$ =30	0.96320		
	OF^{3}_{4}		**29**	**0.96360**
Iteration Solution			**130**	**3.69616**
Fourth Iteration, OF^{4}	$OF^{4}_{(1,1),I}$ =11	0.72400		
	$OF^{4}_{(1,2),I}$ =6	0.72376		
	$OF^{4}_{(1,3),I}$ =9	0.72400		
	$OF^{4}_{1,I}$ =36	0.72352		
	OF^{4}_{1}		**35**	**0.72400**
	$OF^{4}_{(2,1),I}$ =	---------		
	$OF^{4}_{(2,2),I}$ =10	0.88352		
	$OF^{4}_{(2,3),I}$ =7	0.88360		
	$OF^{4}_{2,I}$ =28	0.88336		
	OF^{4}_{2}		**27**	**0.88384**
	$OF^{4}_{(3,1),I}$ =7	1.12416		
	$OF^{4}_{(3,2),I}$ =11	1.12448		
	$OF^{4}_{(3,3),I}$ =8	1.12448		
	$OF^{4}_{3,I}$ =40	1.12472		
	OF^{4}_{3}		**39**	**1.12472**
	$OF^{4}_{(4,1),I}$ =9	0.88352		
	$OF^{4}_{(4,2),I}$ =11	0.96360		
	$OF^{4}_{(4,3),I}$	---------		
	$OF^{4}_{4,I}$ =30	0.96320		

	OF_4^4		**30**	**0.88352**
Iteration Solution			**131**	**3.61608**
Fifth Iteration, OF^5	$OF^5_{(1,1),I}$ =11	0.72400		
	$OF^5_{(1,2),I}$ =6	0.72376		
	$OF^5_{(1,3),I}$ =9	0.72400		
	$OF^5_{1,I}$ =36	0.72352		
	OF_1^5		**35**	**0.72400**
	$OF^5_{(2,1),I}$ =	---------		
	$OF^5_{(2,2),I}$ =10	0.88352		
	$OF^5_{(2,3),I}$ =7	0.88360		
	$OF^5_{2,I}$ =28	0.88336		
	OF_2^5		**27**	**0.88384**
	$OF^5_{(3,1),I}$ =7	1.12416		
	$OF^5_{(3,2),I}$ =11	1.12448		
	$OF^5_{(3,3),I}$ =8	1.12448		
	$OF^5_{3,I}$ =40	1.12472		
	OF_3^5		**40**	**1.12416**
	$OF^5_{(4,1),I}$ =9	0.88352		
	$OF^5_{(4,2),I}$ =11	0.88352		
	$OF^5_{(4,3),I}$	---------		
	$OF^5_{4,I}$ =31	0.88352		
	OF_4^5		**30**	**0.88352**
Iteration Solution			**132**	**3.61552**
Sixth	$OF^6_{(1,1),I}$ =11	0.72400		

Iteration, OF^6	$OF^6_{(1,2),I}=6$	0.72376	
	$OF^6_{(1,3),I}=9$	0.72400	
	$OF^6_{1,I}=36$	0.72352	
	OF^6_1	**35**	**0.72400**
	$OF^6_{(2,1),I}=$	---------	
	$OF^6_{(2,2),I}=10$	0.88352	
	$OF^6_{(2,3),I}=7$	0.88360	
	$OF^6_{2,I}=28$	0.88336	
	OF^6_2	**27**	**0.88384**
	$OF^6_{(3,1),I}=8$	1.12416	
	$OF^6_{(3,2),I}=12$	0.96424	
	$OF^6_{(3,3),I}=9$	1.12392	
	$OF^6_{3,I}=41$	1.04464	
	OF^6_3	**41**	**0.96424**
	$OF^6_{(4,1),I}=10$	0.88352	
	$OF^6_{(4,2),I}=12$	0.88352	
	$OF^6_{(4,3),I}$	---------	
	$OF^6_{4,I}=32$	0.88352	
	OF^6_4	**30**	**0.88352**
Iteration Solution		**133**	**3.45560**
Seventh Iteration, OF^7	$OF^7_{(1,1),I}=11$	0.72400	
	$OF^7_{(1,2),I}=6$	0.72376	
	$OF^7_{(1,3),I}=9$	0.72400	
	$OF^7_{1,I}=36$	0.72352	

	OF_1^7		**36**	**0.72352**
	$OF_{(2,1),I}^7 =$	---------		
	$OF_{(2,2),I}^7 = 10$	0.88352		
	$OF_{(2,3),I}^7 = 7$	0.88360		
	$OF_{2,I}^7 = 28$	0.88336		
	OF_2^7		**27**	**0.88384**
	$OF_{(3,1),I}^7 = 9$	0.96424		
	$OF_{(3,2),I}^7 = 13$	0.96424		
	$OF_{(3,3),I}^7 = 10$	0.96400		
	$OF_{3,I}^7 = 42$	0.96424		
	OF_3^7		**41**	**0.96424**
	$OF_{(4,1),I}^7 = 11$	0.88352		
	$OF_{(4,2),I}^7 = 13$	0.88352		
	$OF_{(4,3),I}^7$	---------		
	$OF_{4,I}^7 = 33$	0.88352		
	OF_4^7		**30**	**0.88352**
Iteration Solution			**134**	**3.45512**
Eighth Iteration, OF^8	$OF_{(1,1),I}^8 = 12$	0.72352		
	$OF_{(1,2),I}^8 = 7$	0.72328		
	$OF_{(1,3),I}^8 = 10$	0.72352		
	$OF_{1,I}^8 = 37$	0.72352		
	OF_1^8		**36**	**0.72352**
	$OF_{(2,1),I}^8 =$	---------		
	$OF_{(2,2),I}^8 = 10$	0.88352		

	$OF^{8}_{(2,3),I}=7$	0.88360		
	$OF^{8}_{2,I}=28$	0.88336		
	OF^{8}_{2}		**28**	**0.88336**
	$OF^{8}_{(3,1),I}=9$	0.96424		
	$OF^{8}_{(3,2),I}=13$	0.96424		
	$OF^{8}_{(3,3),I}=10$	0.96400		
	$OF^{8}_{3,I}=42$	0.96424		
	OF^{8}_{3}		**41**	**0.96424**
	$OF^{8}_{(4,1),I}=11$	---------		
	$OF^{8}_{(4,2),I}=13$	---------		
	$OF^{8}_{(4,3),I}$	---------		
	$OF^{8}_{4,I}=33$	---------		
	OF^{8}_{4}		**30**	**0.88352**
Iteration Solution			**135**	**3.45464**
Ninth Iteration, OF^{9}	$OF^{9}_{(1,1),I}=12$	0.72352		
	$OF^{9}_{(1,2),I}=7$	0.72328		
	$OF^{9}_{(1,3),I}=10$	0.72352		
	$OF^{9}_{1,I}=37$	0.72352		
	OF^{9}_{1}		**36**	**0.72352**
	$OF^{9}_{(2,1),I}=$	---------		
	$OF^{9}_{(2,2),I}=11$	0.80352		
	$OF^{9}_{(2,3),I}=8$	0.88312		
	$OF^{9}_{2,I}=29$	0.88336		
	OF^{9}_{2}		**29**	**0.80352**

	$OF^{9}_{(3,1),I}$ =9	0.96424		
	$OF^{9}_{(3,2),I}$ =13	0.96424		
	$OF^{9}_{(3,3),I}$ =10	0.96400		
	$OF^{9}_{3,I}$ =42	0.96424		
	OF^{9}_{3}		**41**	**0.96424**
	$OF^{9}_{(4,1),I}$ =11	---------		
	$OF^{9}_{(4,2),I}$ =13	---------		
	$OF^{9}_{(4,3),I}$	---------		
	$OF^{9}_{4,I}$ =33	---------		
	OF^{9}_{4}		**30**	**0.88352**
Iteration Solution			**136**	**3.37480**
Tenth Iteration, OF^{10}	$OF^{10}_{(1,1),I}$ =12	0.72352		
	$OF^{10}_{(1,2),I}$ =7	0.72328		
	$OF^{10}_{(1,3),I}$ =10	0.72352		
	$OF^{10}_{1,I}$ =37	0.72352		
	OF^{10}_{1}		**37**	**0.72328**
	$OF^{10}_{(2,1),I}$ =	---------		
	$OF^{10}_{(2,2),I}$ =12	0.80352		
	$OF^{10}_{(2,3),I}$ =9	0.80328		
	$OF^{10}_{2,I}$ =30	0.80336		
	OF^{10}_{2}		**29**	**0.80352**
	$OF^{10}_{(3,1),I}$ =9	0.96424		
	$OF^{10}_{(3,2),I}$ =13	0.96424		
	$OF^{10}_{(3,3),I}$ =10	0.96400		

	$OF^{10}_{3,I}$ =42	0.96424		
	OF^{10}_{3}		**41**	**0.96424**
	$OF^{10}_{(4,1),I}$ =11	---------		
	$OF^{10}_{(4,2),I}$ =13	---------		
	$OF^{10}_{(4,3),I}$	---------		
	$OF^{10}_{4,I}$ =33	---------		
	OF^{10}_{4}		**30**	**0.88352**
Iteration Solution			**137**	**3.37456**
	$OF^{11}_{(1,1),I}$ =13	0.72328		
	$OF^{11}_{(1,2),I}$ =8	0.72328		
	$OF^{11}_{(1,3),I}$ =11	0.72328		
	$OF^{11}_{1,I}$ =38	0.72328		
	OF^{11}_{1}		**37**	**0.72328**
	$OF^{11}_{(2,1),I}$ =	---------		
	$OF^{11}_{(2,2),I}$ =12	0.80352		
Eleventh Iteration, OF^{11}	$OF^{11}_{(2,3),I}$ =9	0.80328		
	$OF^{11}_{2,I}$ =30	0.80336		
	OF^{11}_{2}		**30**	**0.80328**
	$OF^{11}_{(3,1),I}$ =9	0.96424		
	$OF^{11}_{(3,2),I}$ =13	0.96424		
	$OF^{11}_{(3,3),I}$ =10	0.96400		
	$OF^{11}_{3,I}$ =42	0.96424		
	OF^{11}_{3}		**41**	**0.96424**
	$OF^{11}_{(4,1),I}$ =11	---------		

	$OF^{11}_{(4,2),I}$ =13	---------		
	$OF^{11}_{(4,3),I}$	---------		
	$OF^{11}_{4,I}$ =33	---------		
	OF^{11}_{4}		**30**	**0.88352**
Iteration Solution			**138**	**3.37432**
Twelfth Iteration, OF^{12}	$OF^{12}_{(1,1),I}$ =14	0.72328		
	$OF^{12}_{(1,2),I}$ =9	0.72328		
	$OF^{12}_{(1,3),I}$ =12	0.72328		
	$OF^{12}_{1,I}$ =39	0.72328		
	OF^{12}_{1}		**37**	**0.72328**
	$OF^{12}_{(2,1),I}$ =	---------		
	$OF^{12}_{(2,2),I}$ =13	0.80328		
	$OF^{12}_{(2,3),I}$ =10	0.80328		
	$OF^{12}_{2,I}$ =31	0.80328		
	OF^{12}_{2}		**30**	**0.80328**
	$OF^{12}_{(3,1),I}$ =9	0.96424		
	$OF^{12}_{(3,2),I}$ =13	0.96424		
	$OF^{12}_{(3,3),I}$ =10	0.96400		
	$OF^{12}_{3,I}$ =42	0.96424		
	OF^{12}_{3}		**42**	**0.96400**
	$OF^{12}_{(4,1),I}$ =11	---------		
	$OF^{12}_{(4,2),I}$ =13	---------		
	$OF^{12}_{(4,3),I}$	---------		
	$OF^{12}_{4,I}$ =33	---------		

	OF_4^{12}		**30**	**0.88352**
Iteration Solution			**139**	**3.37408**
Thirteenth Iteration, OF^{13}	$OF_{(1,1),I}^{13}=15$	0.72328		
	$OF_{(1,2),I}^{13}=10$	0.72328		
	$OF_{(1,3),I}^{13}=13$	0.72328		
	$OF_{1,I}^{13}=40$	0.72328		
	OF_1^{13}		**37**	**0.72328**
	$OF_{(2,1),I}^{13}=$	---------		
	$OF_{(2,2),I}^{13}=14$	0.80328		
	$OF_{(2,3),I}^{13}=11$	0.80328		
	$OF_{2,I}^{13}=32$	0.80328		
	OF_2^{13}		**30**	**0.80328**
	$OF_{(3,1),I}^{13}=10$	0.96400		
	$OF_{(3,2),I}^{13}=14$	0.96400		
	$OF_{(3,3),I}^{13}=11$	0.96400		
	$OF_{3,I}^{13}=43$	0.96400		
	OF_3^{13}		**42**	**0.96400**
	$OF_{(4,1),I}^{13}=11$	---------		
	$OF_{(4,2),I}^{13}=13$	---------		
	$OF_{(4,3),I}^{13}$	---------		
	$OF_{4,I}^{13}=33$	---------		
	OF_4^{13}		**30**	**0.88352**
Iteration Solution			**139**	**3.37408**
Fourteenth	$OF_{(1,1),I}^{14}=15$	---------		

Iteration, OF^{14}	$OF^{14}_{(1,2),I}$ =10	---------		
	$OF^{14}_{(1,3),I}$ =13	---------		
	$OF^{14}_{1,I}$ =40	---------		
	OF^{14}_{1}		**37**	**0.72328**
	$OF^{14}_{(2,1),I}$ =	---------		
	$OF^{14}_{(2,2),I}$ =15	0.80328		
	$OF^{14}_{(2,3),I}$ =12	0.80328		
	$OF^{14}_{2,I}$ =33	0.80328		
	OF^{14}_{2}		**30**	**0.80328**
	$OF^{14}_{(3,1),I}$ =11	0.96400		
	$OF^{14}_{(3,2),I}$ =15	0.96400		
	$OF^{14}_{(3,3),I}$ =12	0.96400		
	$OF^{14}_{3,I}$ =44	0.96400		
	OF^{14}_{3}		**42**	**0.96400**
	$OF^{14}_{(4,1),I}$ =11	---------		
	$OF^{14}_{(4,2),I}$ =13	---------		
	$OF^{14}_{(4,3),I}$	---------		
	$OF^{14}_{4,I}$ =33	---------		
	OF^{14}_{4}		**30**	**0.88352**
Iteration Solution			**139**	**3.37408**
Fifteenth Iteration, OF^{15}	$OF^{15}_{(1,1),I}$ =15	---------		
	$OF^{15}_{(1,2),I}$ =10	---------		
	$OF^{15}_{(1,3),I}$ =13	---------		
	$OF^{15}_{1,I}$ =40	---------		

	OF_{1}^{15}		**37**	**0.72328**
	$OF_{(2,1),I}^{15}$ =	---------		
	$OF_{(2,2),I}^{15}$ =15	---------		
	$OF_{(2,3),I}^{15}$ =12	---------		
	$OF_{2,I}^{15}$ =33	---------		
	OF_{2}^{15}		**30**	**0.80328**
	$OF_{(3,1),I}^{15}$ =12	0.96400		
	$OF_{(3,2),I}^{15}$ =16	0.96400		
	$OF_{(3,3),I}^{15}$ =13	0.96400		
	$OF_{3,I}^{15}$ =45	0.96400		
	OF_{3}^{15}		**42**	**0.96400**
	$OF_{(4,1),I}^{15}$ =11	---------		
	$OF_{(4,2),I}^{15}$ =13	---------		
	$OF_{(4,3),I}^{15}$	---------		
	$OF_{4,I}^{15}$ =33	---------		
	OF_{4}^{15}		**30**	**0.88352**
Iteration Solution			**139**	**3.37408**
Final Iteration	$OF_{(1,1),I}^{16}$ =15	---------		
	$OF_{(1,2),I}^{16}$ =10	---------		
	$OF_{(1,3),I}^{16}$ =13	---------		
	$OF_{1,I}^{16}$ =40	---------		
	OF_{1}^{16}		**37**	**0.72328**
	$OF_{(2,1),I}^{16}$ =	---------		
	$OF_{(2,2),I}^{16}$ =15	---------		

	$OF^{16}_{(2,3),I}=12$	---------		
	$OF^{16}_{2,I}=33$	---------		
	OF^{16}_{2}		**30**	**0.80328**
	$OF^{16}_{(3,1),I}=12$	---------		
	$OF^{16}_{(3,2),I}=16$	---------		
	$OF^{16}_{(3,3),I}=13$	---------		
	$OF^{16}_{3,I}=45$	---------		
	OF^{16}_{3}		**42**	**0.96400**
	$OF^{16}_{(4,1),I}=11$	---------		
	$OF^{16}_{(4,2),I}=13$	---------		
	$OF^{16}_{(4,3),I}$	---------		
	$OF^{16}_{4,I}=33$	---------		
	OF^{16}_{4}		**30**	**0.88352**
Iteration Solution			**139**	**3.37408**

Table 4-5: First heuristic method's solution for Example 5

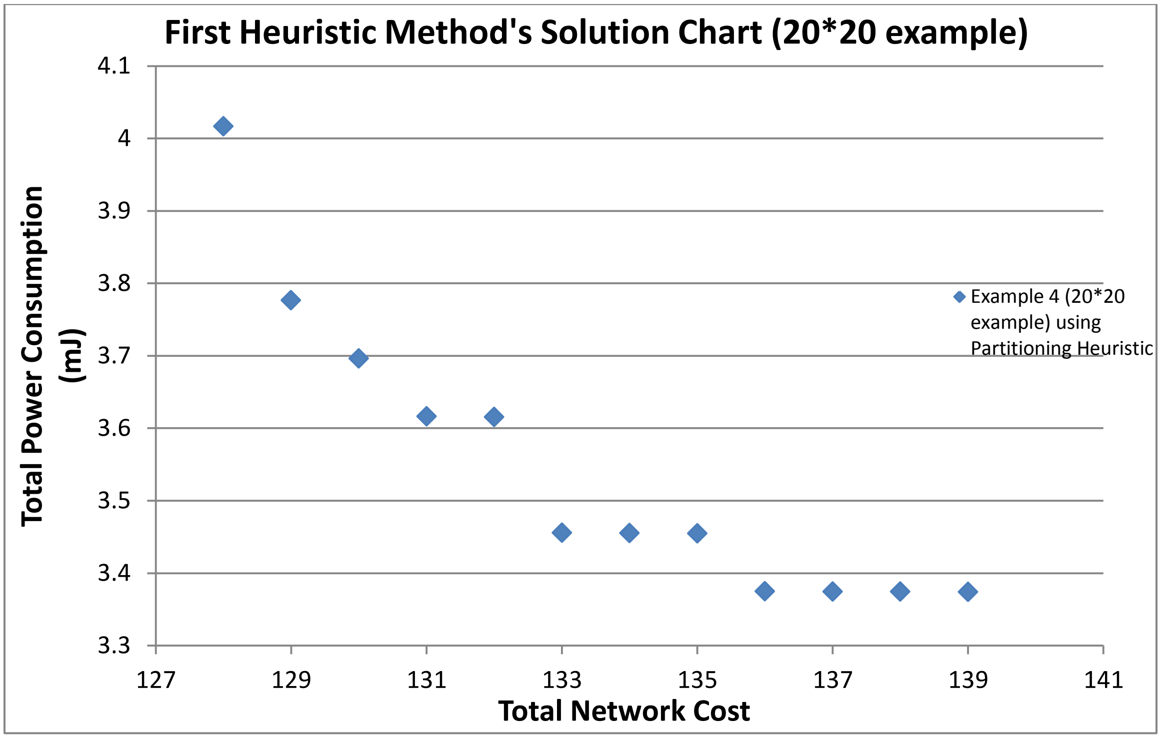

Figure 4-14: First heuristic method's performance for Example 5

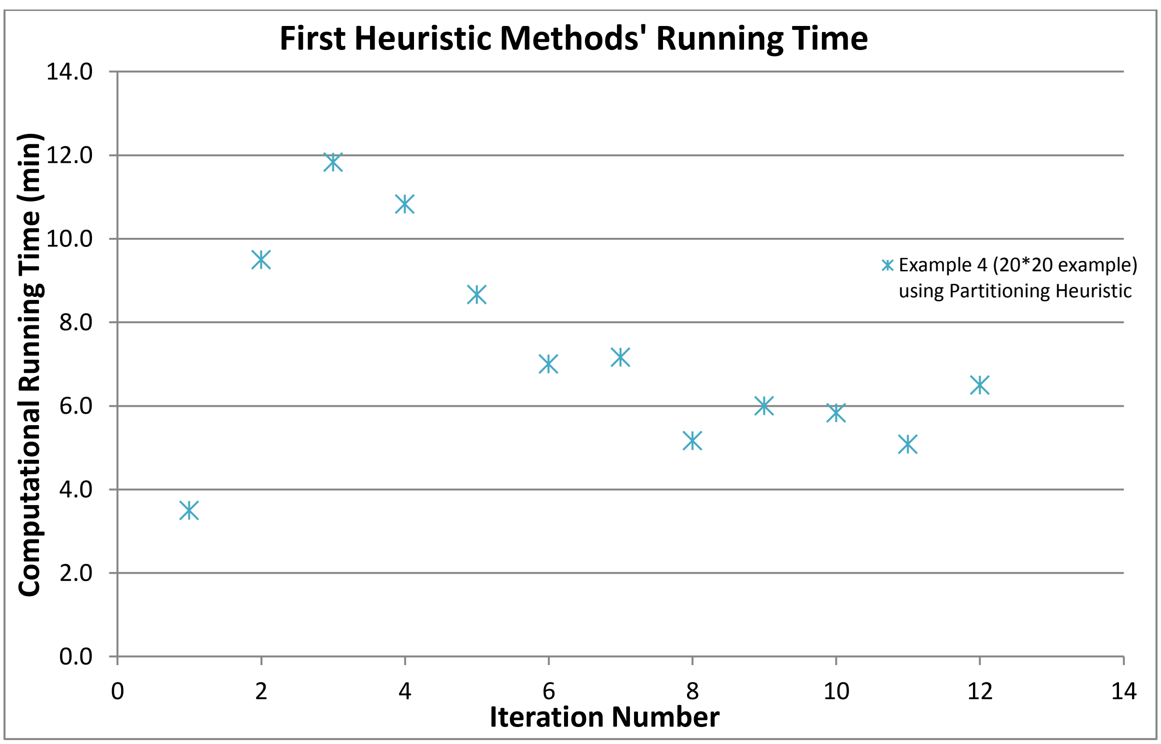

Figure 4-15: First heuristic method's running time for Example 5

4.8 NUMERICAL EXAMPLE 6: 3-DIMENSIONAL SPACE

Given a 3D field of $10\times10\times5$ unit length dimensions, the processing node is fixed at the location $(5,5,0)$. Five critical points need to be covered with known levels of criticality. The critical points' location and criticality are given in the following table.

Critical Point, i	1	2	3	4	5
Location Coordinates, (x_i, y_i, z_i)	(0,0,0)	(10,0,0)	(0,10,0)	(10,10,0)	(5,5,5)
Criticality, D_i	2	1	1	1	1

Table 4-6: Critical points' locations and criticalities for Example 6

We assume the availability of two sensors and two relays in the market. Sensors and relays types and characteristics are the same as in Table 3-2 and Table 3-3 presented previously in Example 1.

The example is solved using the initial First Heuristic method as discussed in Algorithm B previously. MatLab software is used to graph the space solution for all subsequent iterations.

Iteration	First Space Partitioning Heuristic Solution		Iteration Solution	
	OF_I = Network Cost ($)	OF_{II} = Total Energy Consumption (mJ)	OF_I = Network Cost ($)	OF_{II} = Total Energy Consumption (mJ)
First Iteration, OF^1	$OF^*_{1,I}$: 10	0.32144	10	0.32144
	$OF^*_{2,I}$: 8	0.24128	8	0.24128
	$OF^*_{3,I}$: 8	0.24128	8	0.24128
	$OF^*_{4,I}$: 8	0.24128	8	0.24128
	$OF^*_{5,I}$: 5	0.16040	5	0.16040
Iteration Solution			**39**	**1.20568**
Second Iteration, OF^2	11	0.32136	10	0.32144
	9	0.24112	9	0.24112
	9	0.24112	8	0.24128
	9	0.24112	8	0.24128
	6	0.16040	5	0.16040
Iteration Solution			**40**	**1.20552**
Third Iteration, OF^3	11	0.32136	10	0.32144
	10	0.24112	9	0.24112
	9	0.24112	9	0.24112
	9	0.24112	8	0.24128
	7	0.16040	5	0.16040
Iteration Solution			**41**	**1.20536**
Fourth Iteration, OF^4	11	0.32136	10	0.32144
	11	0.24112	9	0.24112
	10	0.24112	9	0.24112
	9	0.24112	9	0.24112
	8	0.16040	5	0.16040
Iteration Solution			**42**	**1.20520**

Fifth Iteration, OF^5	11	0.32136	11	0.32136
	12	0.24112	9	0.24112
	11	0.24112	9	0.24112
	10	0.24112	9	0.24112
	----------	-----------	5	0.16040
Iteration Solution			**43**	**1.20512**
Sixth Iteration, OF^6	12	0.32120	12	0.32120
	----------	-----------	9	0.24112
	12	0.24112	9	0.24112
	11	0.24112	9	0.24112
	----------	-----------	5	0.16040
Iteration Solution			**44**	**1.20496**
Seventh Iteration, OF^7	13	0.32120	12	0.32120
	----------	-----------	9	0.24112
	----------	-----------	9	0.24112
	12	0.24112	9	0.24112
	----------	-----------	5	0.16040
Iteration Solution			**44**	**1.20496**
Eighth Iteration, OF^8	14	0.32120	12	0.32120
	----------	-----------	9	0.24112
	----------	-----------	9	0.24112
	----------	-----------	9	0.24112
	----------	-----------	5	0.16040
Iteration Solution			**44**	**1.20496**
Ninth Iteration, OF^9	15	0.32120	12	0.32120
	----------	-----------	9	0.24112
	----------	-----------	9	0.24112
	----------	-----------	9	0.24112
	----------	-----------	5	0.16040
Iteration Solution			**44**	**1.20496**
Final Solution	----------	-----------	12	0.32120

	----------	-----------	9	0.24112
	----------	-----------	9	0.24112
	----------	-----------	9	0.24112
	----------	-----------	5	0.16040
Iteration Solution			**44**	**1.20496**

Table 4-7: First heuristic method's solution for Example 6

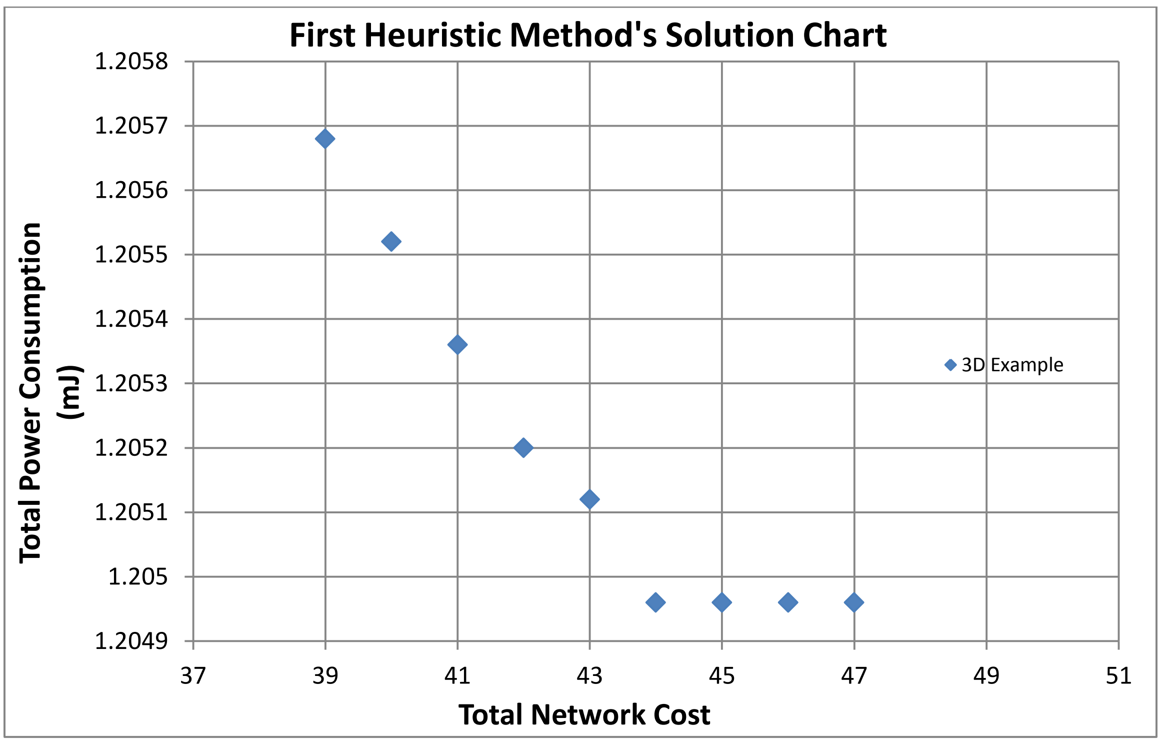

Figure 4-16: First heuristic method's performance for Example 6

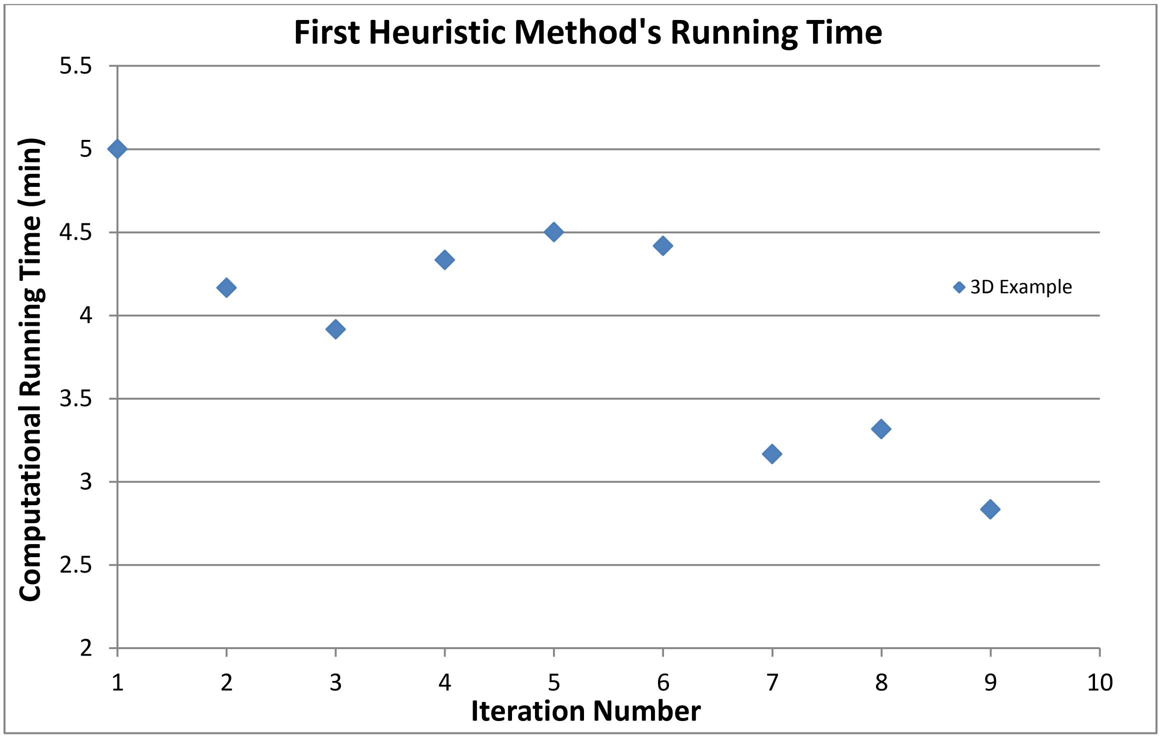

Figure 4-17: First heuristic method's running time for Example 6

The following pages will represent graphical illustrations of the 3D space solutions for all the iterations listed in the above table. The critical points, sensors and relays will be shown clearly. Likewise, the solution (sensors types and locations, relays types and locations and transmission links) will be clearly graphed for all iterations.

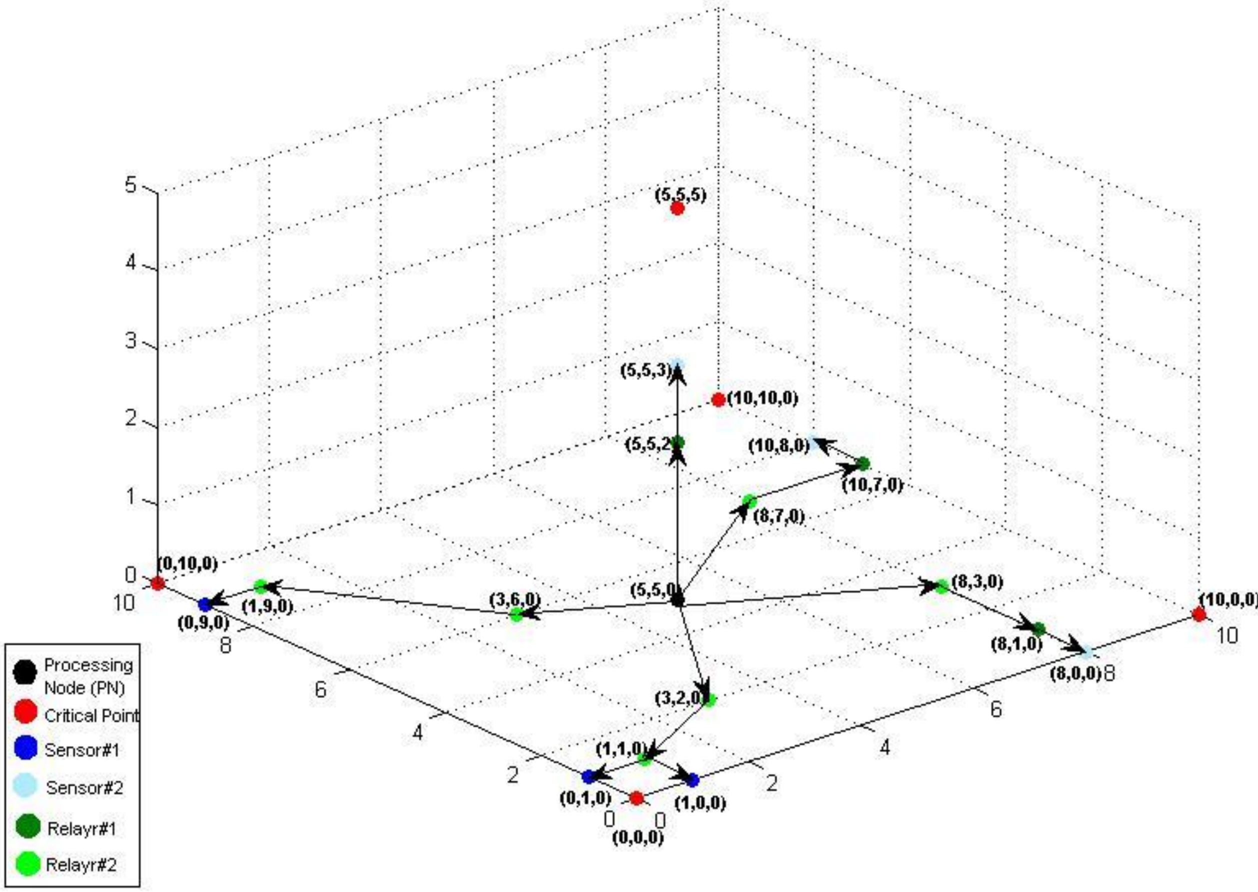

Figure 4-18: Model I solution for Iteration 1

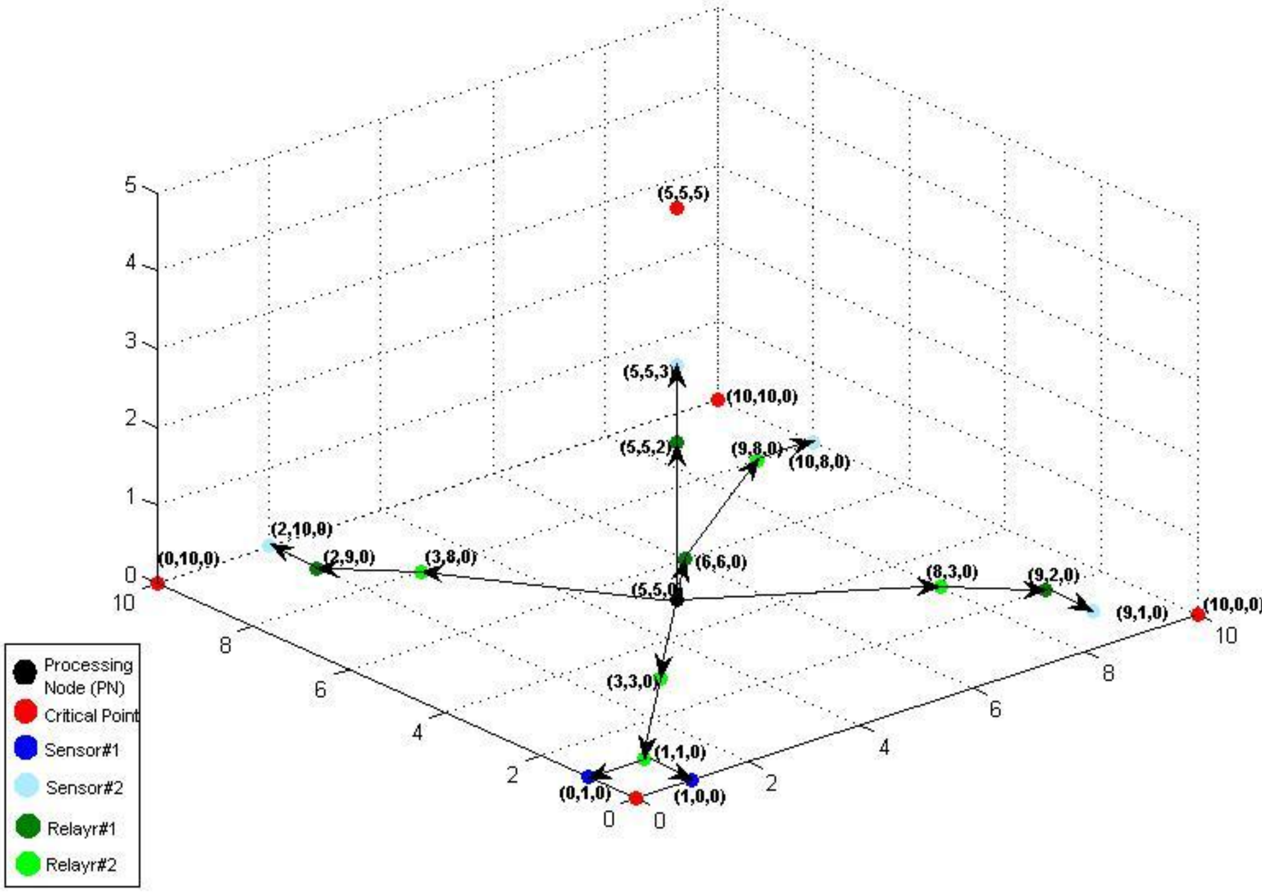

Figure 4-19: Bi-objective solution for Iteration 1

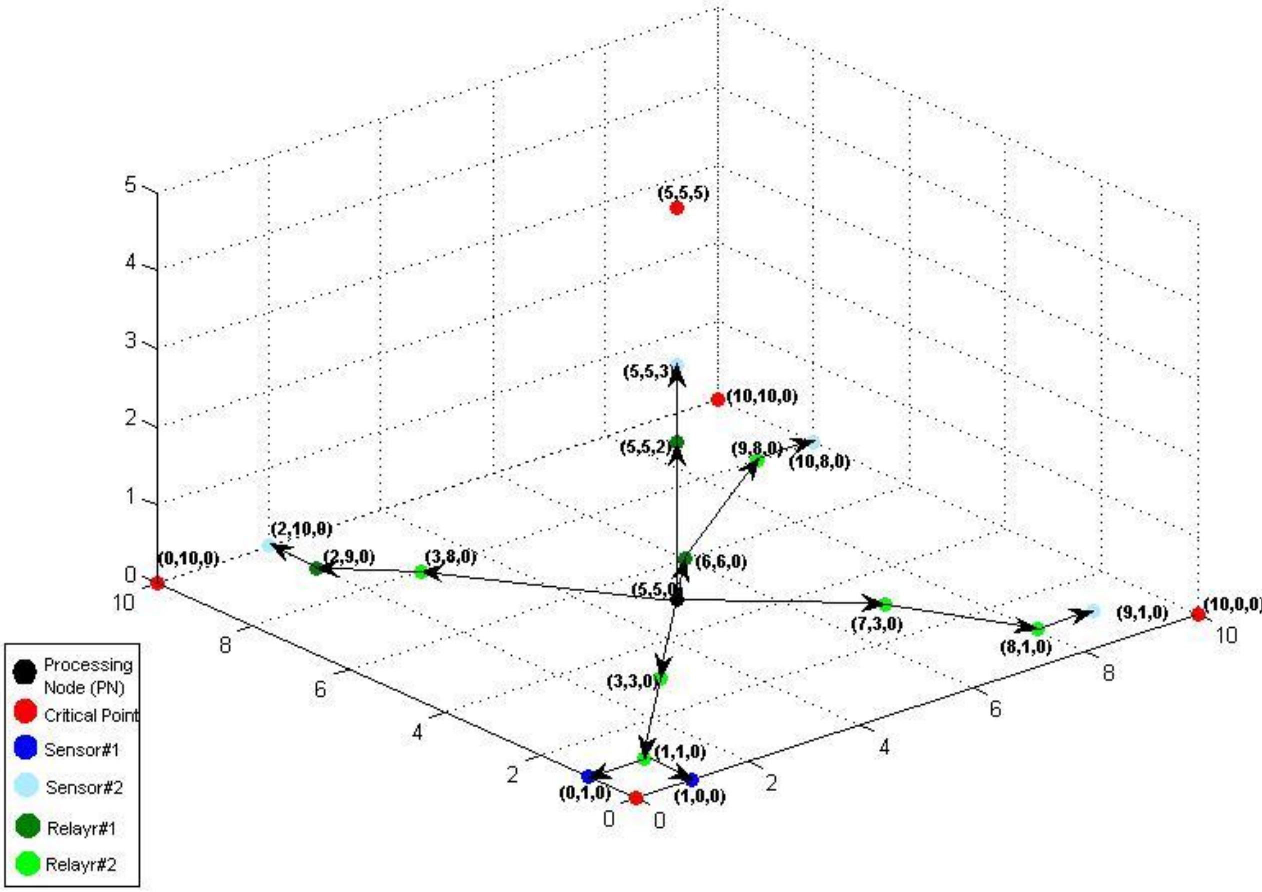

Figure 4-20: Bi-objective solution for Iteration 2

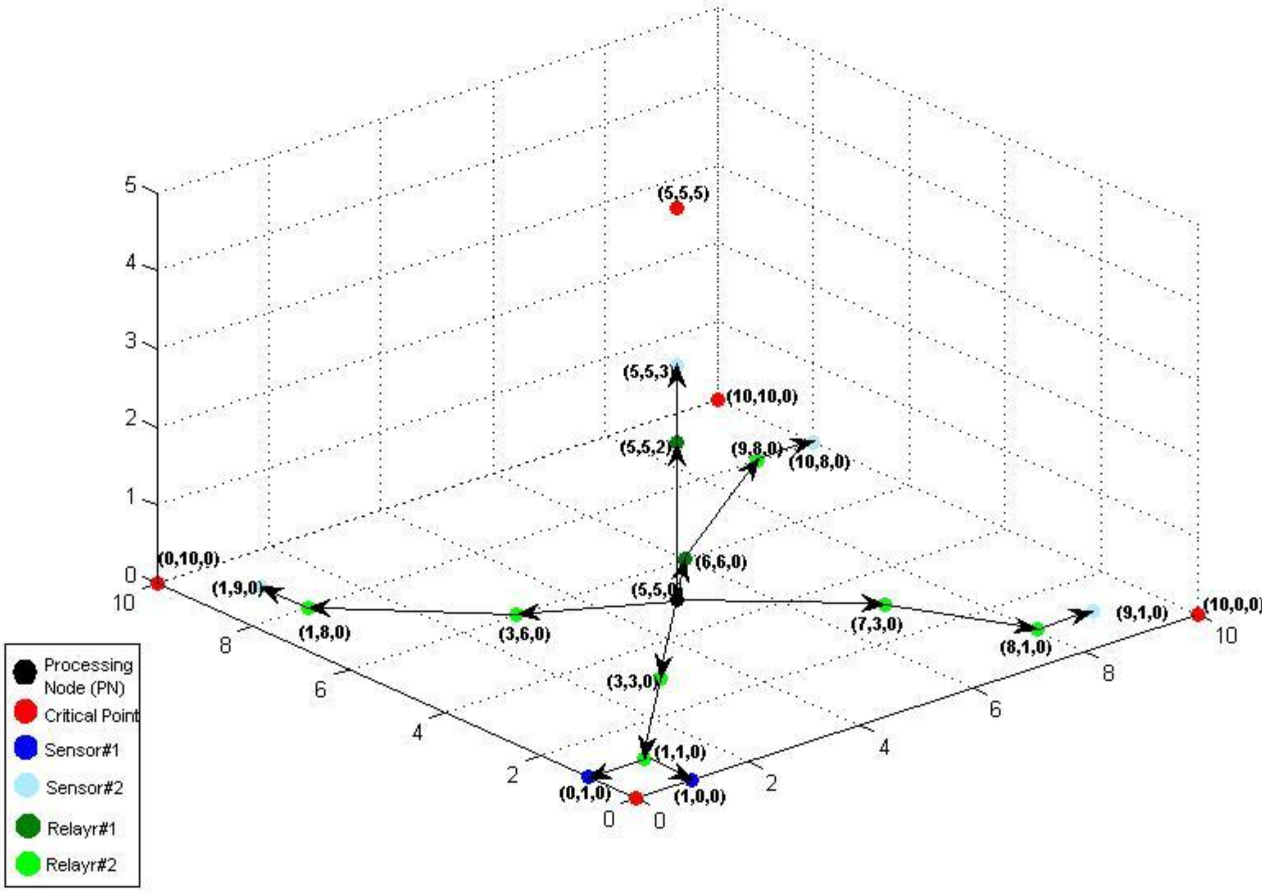

Figure 4-21: Bi-objective solution for Iteration 3

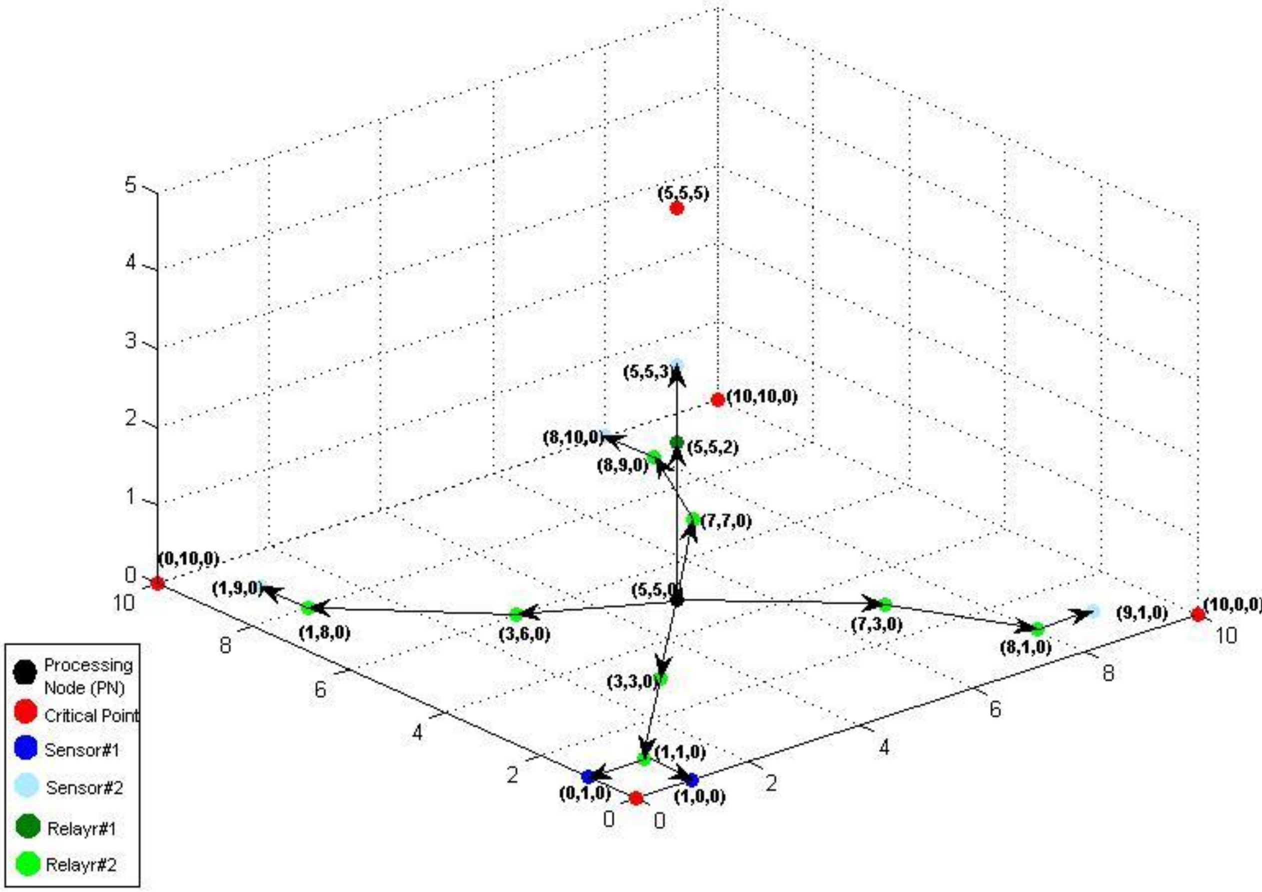

Figure 4-22: Bi-objective solution for Iteration 4

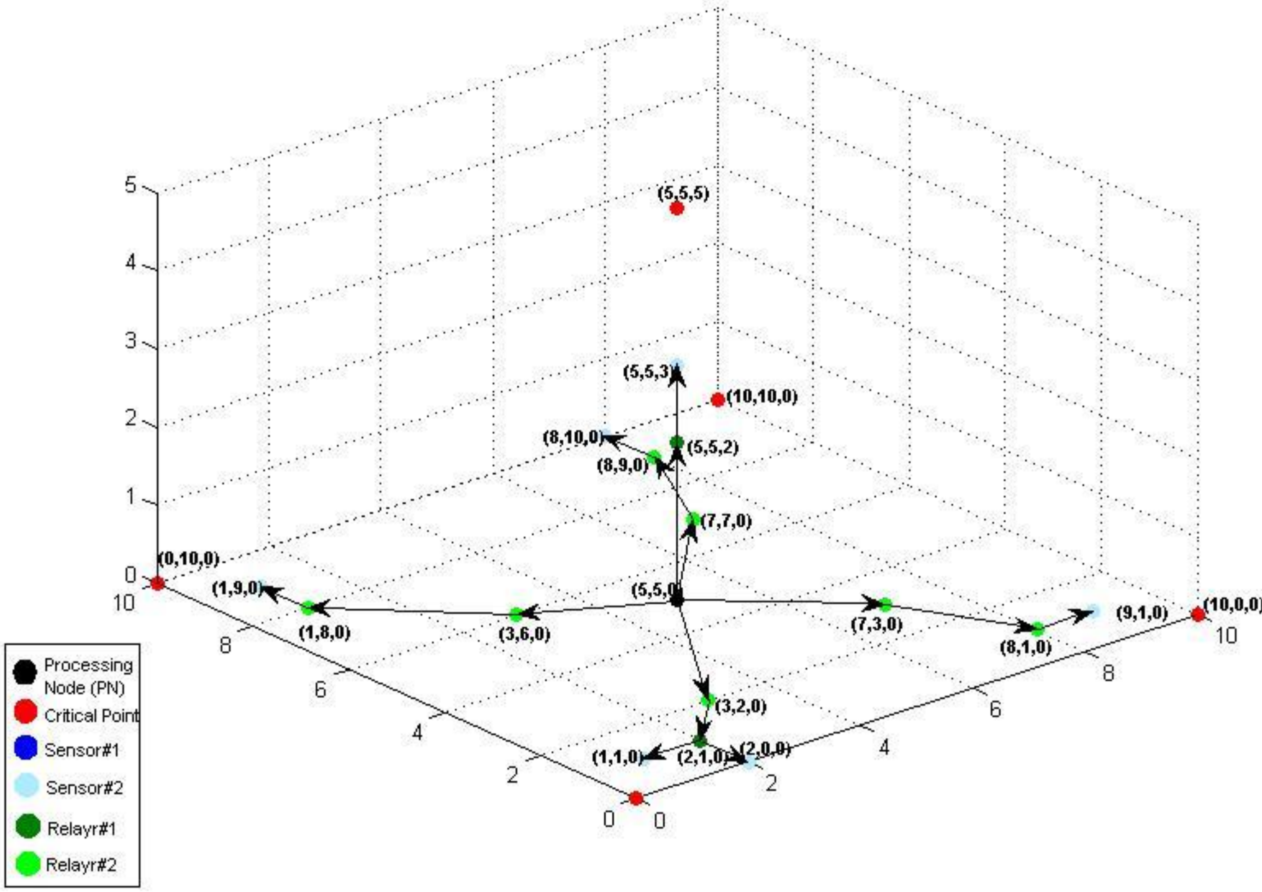

Figure 4-23: Bi-objective solution for Iteration 5

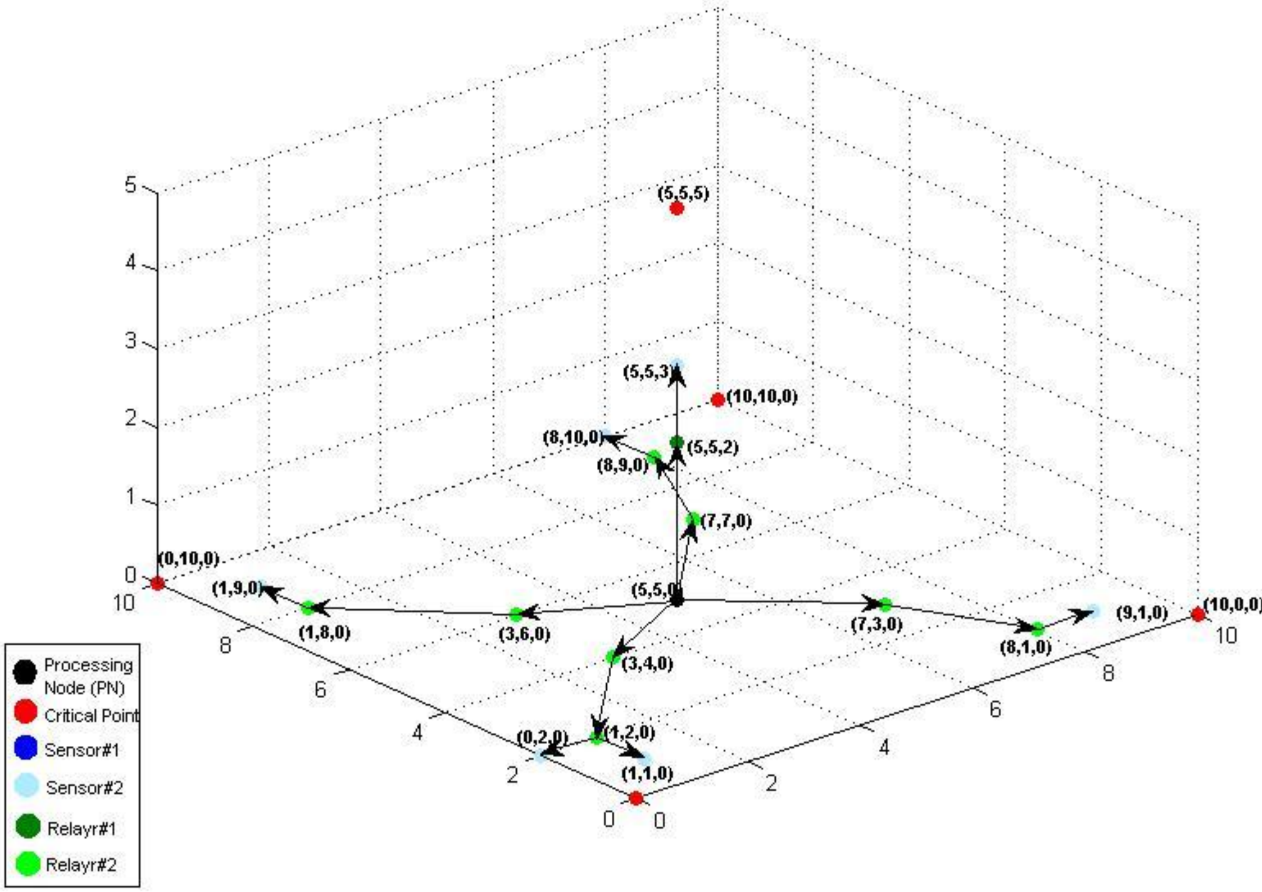

Figure 4-24: Bi-objective solution for Iteration 6

The graphical solution shows clearly the effect of increasing the cost on the network design. As expected, the number of sensors and relays will increase as the cost limit, i.e. budget, does. An inevitable result as the cost increases is that sensors will be placed as far as possible from the critical points. This is a reasonable trend since there is no power consumption associated with the sensing range. Thus the sensors will attempt to locate as far as they can from the critical points and try to minimize the distance to the processing node.

Chapter 5

CONCLUSION

5.1 PREFACE

This chapter concludes the work of the thesis. A review of the designed Integer Linear Program and the proposed heuristic method are presented. Finally, possible extensions of the thesis work are recommended.

In this thesis, an Integer Linear Program (ILP) was developed to solve the problem of locating sensors and relays in a bounded space. The space could be a two or three dimensional bounded facility. The literature review has shown that no attempt has been made to solve this problem in the same way tackled in this thesis. The powerfulness of this work underlies in the comprehensive detailed solution it provides to the user. The user is only requested to insert the locations of the critical points and their relative criticality. The ILP will provide the user with the type and location for each sensor and relay to be placed in the field and the transmission path to the processing node for each deployed sensor.

For an attempt to overcome the exponential increasing time for the ILP, a Space Partitioning Heuristic Method was developed. The heuristic provides near optimal solutions in relatively much lower computational time. The heuristic method is based on partitioning the bounded space into smaller parts and solving each partition independently using the proposed ILP model. Then the solutions for all the parts are combined together to constitute the overall space solution. Yet some guidelines and suggestions have been listed to direct the user in partitioning the space, the space partitioning step is generally ambiguous and tedious to control optimally. As a general hint, the user should make an attempt to include the processing node in each formed partition. This will eliminate the need for assuming virtual processing node, to guarantee connectivity. For large spaces, it may not be acceptable, in terms of partition size, to include the processing node in some partitions. In this case, a partition containing the critical point is formed and a virtual processing node is located inside that partition. The virtual processing node should be placed at the grid point which has the minimum Euclidean distance from the actual processing node.

For future studies and extensions of this work, the researcher can consider the random shaped ranges (i.e. sensing or transmission ranges). It is known that in practice, the sensing or transmission range is, to the best case, a precise estimate of the actual range and some randomness exists. This nature of the range can be described as a distribution with known parameters and fed to the model for better decision making.

As a second extension work, the researcher can consider the location of the processing node to be a decision variable. This will push the model to find the optimal location for the processing node that optimizes the objective function.

Also, some extended analysis could be conducted on the optimal or upper bound dimensions of the space. This will help the users to specify the partition size ahead of time with more confidence and accuracy.

REFERENCES

[1] K. Chakrabarty, S. Sitharama Iyengar and H. Qi, "Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks", IEEE Transactions on Computers, vol. 51, no. 12, pp. 1448-1453, December 2002.

[2] A. Bari, D. Teng and A. Jaekel, "Optimal Relay Node Placement in Hierarchical Sensor Networks with Mobile Data Collector", IEEE Computer Society, pp.1–6, 2009.

[3] A. Chamam and S. Pierre, "On the Planning of Wireless Sensor Networks: Energy-Efficient Clustering under the Joint Routing and Coverage Constraint", IEEE Transactions On Mobile Computing, vol. 8, no. 8, pp. 1077-1086, August 2008.

[4] S. S. Dhillon and K. Chakrabarty, "Sensor Placement for Effective Coverage and Surveillance in Distributed Sensor Networks", Wireless Communications and Networking, vol. 3, no. 8, pp. 1609-1614, 2003.

[5] J. Wang and N. Zhong, "Efficient Point Coverage in Wireless Sensor Networks", Journal of Combinatorial Optimization, vol. 11, no. 3, pp. 291-304, 2006.

[6] Waleed Alsalih, Hossam Hassanein, and Selim Akl, "Placement of multiple mobile data collectors in underwater acoustic sensor networks", Wireless Communications and Mobile Computing, vol. 1, no. 13, 2007.

[7] Enrico Natalizio, Valeria Loscr´ı, and Valeria Loscr´ı, “Optimal Placement of Wireless Nodes for Maximizing Path Lifetime”, IEEE Communications Letters, vol. 12, no. 5, May 2008.

[8] Quanhong Wang, Kenan Xu, Glen Takahara, and Hossam Hassanein, “Device Placement for Heterogeneous Wireless Sensor Networks: Minimum Cost with Lifetime Constraints”, IEEE Transactions on Wireless Communications, vol. 6, no. 7, July 2007.

[9] Laurence A. Wolsey and George L. Nemhauser, Integer and Combinatorial Optimization, John Wiley, 1999.

[10] Maulin Patel, R. Chandrasekaran and S.Venkatesan, “Energy Efficient Sensor, Relay and Base Station Placements for Coverage, Connectivity and Routing”, Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International, pp. 581-586, April 2005.

[11] Xingzhen Bai, Shu Li and Juan Xu, “Mobile Sensor Deployment Optimization for k-Coverage in Wireless Sensor Networks with a Limited Mobility Model”, IETE TECHNICAL REVIEW, vol. 27, no. 2, pp. 124-137, March-April 2010.

VITA

Name: Tamer Mohamed Deyab.

Current Address: King Fahd University of Petroleum & Minerals – Dhahran 31261 – Saudi Arabia.

Permanent Address: King Fahd University of Petroleum & Minerals – Dhahran 31261 – Saudi Arabia.

Telephone: (+966)50-140-94-26

Email: tamer.deyab@gmail.com

Education:

1- M.Sc. in Systems Engineering (Industrial Engineering and Operations Research), King Fahd University of Petroleum and Minerals, Dhahran, K.S.A, (03/2009 – 06/2009).

2- B.Sc. in Industrial and Systems Engineering (ISE), King Fahd University of Petroleum and Minerals, Dhahran, K.S.A, (03/2004 – 02/2009).